Cloud BI: A Multi-party Authentication Framework for Securing Business Intelligence on the Cloud

UNIVERSITY OF DERBY

Cloud BI: A Multi-party Authentication Framework for Securing Business Intelligence on the Cloud

Hussain Al-Aqrabi

Doctor of Philosophy 2016

Table of Contents

List of Figures

List of Tables

Abbreviations

BI	Business Intelligence
CCA	Cloud Certification Authority
Cloud BI	Cloud-based Business Intelligence
DDL	Data Definition Language
DML	Data Manipulation Language
DSS	Decision-support systems
DDOS	Distributed Denial of Service
ES1	Experimental System 1
ES2	Experimental System 2
ETL	Extracting, transforming, and loading
IDPS	Intrusion Detection and Prevention Systems
IPS	Intrusion Prevention System
LDAP	Lightweight directory access protocol
MAS	Multiparty Authentication System
NIST	National Institute of Standards and Technology
OLAP	Online Analytical Processing
OLTP	Online transaction processing
OSCL	Object Security Constraint Language
RADIUS	Remote Authentication Dial-In User Service
RDBMS	Relational Database Management System
SAC	Session Authentication Cloud
SDO	Secure Distributed-OLAP aggregation protocol
SSL	Secured Socket Layers
TACACS	Terminal Access Controller Access Control System
UML	Universal Modelling Language
UTM	Unified Threat Management
XOLAP	XML based OLAP

Acknowledgements

I wish to express sincere appreciation to all those people who made this thesis possible and an unforgettable experience for me.

First of all, I owe debt of gratitude to my PhD advisors Professor Lu Liu and Professor Richard Hill for their contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating. I also remain indebted for their constant support and patience during the difficult time I experienced while presenting the final draft.

I would like to express my deepest gratitude to Prof Nick Antonopoulos, Dr. Victoria Carpenter, James Hardy, Prof. Jianxin Li, and Prof. Zhijun Ding and also to all of the Department faculty members for their help and support.

Finally, I am indebted to my wife Sumaia and our three children for their wonderful support during this study and to my parents, family and friends for their continued encouragement, support and patience.

Abstract

Business intelligence (BI) has emerged as a key technology to be hosted on Cloud computing. BI offers a method to analyse data thereby enabling informed decision making to improve business performance and profitability. However, within the shared domains of Cloud computing, BI is exposed to increased security and privacy threats because an unauthorised user may be able to gain access to highly sensitive, consolidated business information. The business process contains collaborating services and users from multiple Cloud systems in different security realms which need to be engaged dynamically at runtime. If the heterogamous Cloud systems located in different security realms do not have direct authentication relationships then it is technically difficult to enable a secure collaboration. In order to address these security challenges, a new authentication framework is required to establish certain trust relationships among these BI service instances and users by distributing a common session secret to all participants of a session. The author addresses this challenge by designing and implementing a multiparty authentication framework for dynamic secure interactions when members of different security realms want to access services. The framework takes advantage of the trust relationship between session members in different security realms to enable a user to obtain security credentials to access Cloud resources in a remote realm. This mechanism can help Cloud session users authenticate their session membership to improve the authentication processes within multi-party sessions. The correctness of the proposed framework has been verified by using BAN Logics. The performance and the overhead have been evaluated via simulation in a dynamic environment. A prototype authentication system has been designed, implemented and tested based on the proposed framework. The research concludes that the proposed framework and its supporting protocols are an effective functional basis for practical implementation testing, as it achieves good scalability and imposes only minimal performance overhead which is comparable with other state-of-art methods.

Chapter 1: Introduction

1.1 **Research Background**

1.1.1 Business Intelligence on the Clouds

Business intelligence (BI) is a process of extracting analytical knowledge from discovered data clusters created by using data aggregating techniques mixed with domain knowledge for extricating data from various databases. The system comprises of data warehouses, data marts, online analytical processing, and ETL (extracting, transforming, and loading) tools. The BI user dashboards comprise a framework for presentation of analytical knowledge in the form of interactive graphics designed as per the business domain and for solving management and technology problems (Huang and Tseng 2009).

The process of extraction, transformation, and loading is an end-to-end process flow comprising the steps listed below (Huang and Tseng 2009):

(a) Data preparation

(b) Data selection

(c) Data transformation

(d) Data mining

(e) Knowledge discovery

(f) Knowledge identification

(g) Knowledge representation

(h) Knowledge accumulation and storage

(i) Knowledge transfer

(j) Knowledge mining

(k) Knowledge presentation and application

The following figure presents a simple presentation of BI framework in a business organisation. The transactional data is extracted from the transactional databases of the organisation, like SCM, accounting, finance, HR, and marketing. The data is cleaned for eliminating errors, duplicates, inconsistencies, and non-standard forms and transformed as per the structure defined in the data warehouses used for running Online analytical processing (OLAP) reports, data mining for specific usage, and generating mass reports for decision support. The decision maker may use a combination of the three (OLAP, data mining, DSS reporting) for analysing scenarios and making decisions (Giovinazzo 2002).

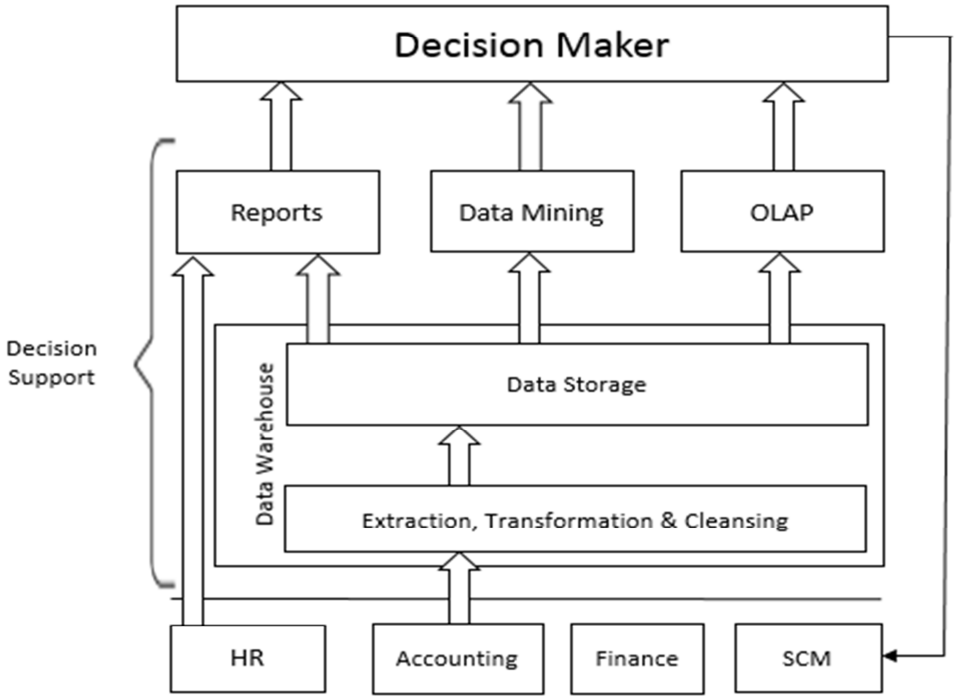

Figure 1: A simple representation of Business Intelligence in a business organisation

Internet enabled business intelligence needs to be deployed on distributed data warehouses with distributed servers and logically integrated data stores (Giovinazzo 2002). BI on distributed systems is achieved using XML based externalisation process achieved through refinement of knowledge following a standard document for standardisation, that

takes five categories of information as inputs, refines them as per information characteristics, XML characteristics, and knowledge model characteristics, and stores them in a pattern needed by management workers. The management workers use data analysis techniques for using the analytical knowledge in decision-making scenarios (Huang and Tseng 2009).

Figure 2: XML based knowledge externalisation process flow (Huang and Tseng 2009)

'content removed for copyright reasons'

The XML based knowledge externalisation process flow presented by Huang and Tseng (2009) employs XML data warehouses and XML based online analytical processing (OLAP) (Mahboubi et al. 2009). XML based data warehouses and OLAP systems can be implemented on web services architecture. The document modelling structures of XML serves as web-based relational databases. The XOLAP (XML based OLAP) can be used to conduct OLAP operations (like roll-up, drill down, and slice and dice) on XML databases. The OLAP cubes can be multi-dimensional XML documents having the attributes at the nodes and facts as branches.

XML based data warehousing and XOLAP can be hosted on web 2.0 architecture on cloud computing. The components needed for implementing BI on cloud computing are web-based workflow engines (like Hadoop or Google App Engine), XML data files stored on cloud based clusters of servers, XOLAP engines, ETL on XML databases, web-based business rules, web-based dashboards, plug ins for browser interfacing, and management interfaces (Wu and Qin 2011). The key advantages of hosting BI on clouds are on-demand self-service, rapid elasticity, resource pooling, and on-demand broadband network access (Stipic and Bronzin 2012).

Web-based BI has more security challenges than BI frameworks hosted on private LANs (Grabova et al. 2011). There are many security challenges in cloud computing

accessed over Internet, and hence BI can be affected by them. This problem is reviewed in the next sub-section.

1.1.2 Cloud Computing Security and Business Intelligence

Cloud computing has some prominent threats that are not prevalent in self-hosted LAN based IT infrastructures. As per Gartner report on cloud security, cloud computing has the following security related threats (Che et al. 2011):

(a) Abuse of resource pools of clouds

(b) Insecure application programming interfacing

(c) Vendor controlled risk management with an unknown risk profile per customer

(d) Identity theft and services hijacking

(e) Insecure boundaries between tenants using the same resources pool

(f) Privacy breaches

(g) Data loss or theft

(h) Protection of user privileges

(i) Segregation of data

(j) Location of data

(k) Inadequate forensics support

(l) Inadequate regulatory compliance

(m) Unclear recovery strategies

(n) Unclear long-term viability

Business intelligence on the cloud can be implemented in the form of a multilayer architecture, having a data layer, a logical layer, and an access layer. The design, construction, and population of data warehouses can be done at the data layer. At the logical,

data can be modelled, managed, and grouped in the form of data marts. All the ETL procedures can be executed at the logical layer. At the access layer, the application interfaces (user and programming) and the middleware can be enabled. Users may access the decision support tools (information mining, OLAP, querying, and reporting) at the access layer. Programmers may access this layer for developing BI applications as per the needs of management workers. This model may be owned by enterprises using cloud hosted BI or service providers offering BI as software as a service (Ouf and Nasr 2011).There can be security threats at each layer of the BI framework in cloud computing.

1.1.3 Research Problem

In modern times, the majority of high value / high profit business industries all are utilize and benefit from Internet based computing. Cloud based systems are a focal technology in this regard that provide utility and ease with regard to universal availability and timely access. However Cloud computing has its own constraints and security considerations that need to be taken into account with regard to its effective application.

Business Intelligence has been identified as a major commercial and technological development that cloud computing can host and enhance. Business Intelligence provides a way of analysing data in a meaningful manner so as to facilitate decision making aimed at increasing productivity and enhancing business performance. Currently, distributed applications in business are encompassing an increasing level of computerisation and a similarly increasing level of enthusiasm. In a multi-tenancy setting, there is dynamism in most cloud based Business Systems which means authentication interfaces should also be dynamic. Nevertheless, in an environment where the domains of Cloud computing are shared, Business Intelligence is more vulnerable to rising security and privacy risks as valuable business information may become more accessible to both hackers and competitors.

Businesses can enrich themselves from the dynamic and flexible cross-institutional service based company procedures. Nevertheless these attributes also bring another dimension in security problems.

Generally a business procedure should be flexible in both application and method to enable dynamic business response. An effect of this is that the execution sequence of a given process may not be predictable for all circumstances even to the extent that sometimes, the real process of execution can be a "one-of-a-kind" (Xu et al. 2012). Thus, the services and applications employed in a procedure are characteristically heterogeneous and might be offered and maintained by various unrelated organizations. Companies have their own homogenous security mechanisms and policies to protect their local resources against security threats but applications residing on the resources of different organizations operate in numerous different heterogeneous *security realms.*

A Security realm may be viewed as a group of principals (like, people, services, and computers) registered with a certain authentication authority (a trusted principal) and controlled through a consistent set of security processes and policies to permit access to services and resources. An authentication authority is a principal that is considered universally trusted and can be relied upon to execute trustworthy authentication functions. As shown by this explanation, authentication is vital for each security realm and before a principal can have a right to use the resources controlled by a security realm, verification of its identity must be confirmed by the authentication procedure of the security realm in order to ascertain the principal who it purports to be. To identify a principal during the process of authentication, the principal has to announce credentials that were provided to it by the authentication authority of the security realm.

In view of the fact that services and organizations can adopt a collaborative process in an extremely vibrant and flexible manner, direct cross-realm authentication relationship is not

simply a means of joining the two collaborating realms. A verification process of the credentials provided by the authentication authority has to be performed in the other security realm where there is a direct cross-realm authentication relationship linking two security realms because of the existence of an interoperable authentication mechanism. A likely solution will be locating some intermediate realms that will connect a pair of separate realms, hence serving as an authentication path for collaborating the pair of different realms. Nevertheless the cost of creating the authentication path for the two disjoint realms is significant as it may involve a lot of other processes for credential conversion that will need extensive invocations to intermediate services.

The lack of authentication path connecting two security realms will necessitate two security realms, when working together, to follow a more traditional and long route that will involve creating a mutual trust entailing entering into contractual agreements, multi-round cooperation and human intervention. Nevertheless it is very hard to achieve a cross-realm authentication in an extremely dynamic service-based business procedure, without negatively impacting service applicability in cross-organizational E-commerce.

The multi-party session oriented authentication protocol enables authenticating service instances that participate in a session to solve the problem. However the protocol provides a commonly shared session secret for all services. Such a group authentication protocol requires sharing of a number of security attributes, e.g. secrets for group session key, secrets for private keys, secrets for key duplication, secrets for session forwarding to other clouds, and secrets for control of keys. In some cases, it is challenging for a session user to determine and verify whether the service instance it contacts is a member of the same multi-party session (Hada and Maruyama 2002). This leads to severe session management vulnerability that could potentially be exploited.

1.2 **Motivation**

Business intelligence with OLAP, data warehousing, and data mining solutions are key requirements for any business in the modern world. Given the dynamics and uncertainties in modern markets, business managers need decision aid systems for making informed decisions. Hence, BI has become a necessity in modern businesses. In addition, BI needs to be Internet enabled for ensuring multi-location access for decision makers (Giovinazzo 2002).

In self-hosted environments it was feared that BI will eventually face a resource crunch situation because it won't be feasible for companies to keep on adding resources to host the never ending expansion of data warehouses and the online analytical processing (OLAP) demands on the underlying networking.

Given the significant advantages of cloud computing (Che et al. 2011), and the deployment readiness of BI using web services architecture (Ouf and Nasr, 2011), the disadvantages of traditional BI can be eliminated by hosting it on cloud computing. As described by Ouf and Nasr (2011), the biggest hurdle in growth of BI has been the cost of its infrastructure and the rapid growth of capacity demands. A business entity need not make capital investments in IT infrastructures and capacity on cloud computing. Clouds offer resources on demand (rapid elasticity, as suggested by (Che et al. 2011), and hence businesses can grow their BI as per the business requirements.

The business systems running on clouds are dynamic and hence the authentication interactions need to be dynamic as well. However, in the shared domains of Cloud computing, BI is exposed to security and privacy threats. This is because an unauthorised user can gain access to highly sensitive consolidated business information in a BI system. The business process contains collaborating services from multiple heterogeneous security

realms which need to be engaged dynamically at runtime, if the security realms do not have direct authentication relationships it is technically difficult to enable secure collaboration. In order to address these security challenges, thus new authentication mechanisms and protocols are required to establish certain trust relationships among these service instances by distributing a common session secret to all participants of a session.

Given this opportunity in deployment and growth of BI on cloud, a number of researchers have contributed to this area. From the literatures on deploying and managing BI on clouds (for example, the three studies by Ouf and Nasr 2011), it was clear that BI on clouds will be very complex. Hence, the security components for securing BI on the clouds must be deployed within complex cloud security architectures. The opportunity was evident given the evolving solutions on securing cloud computing platforms. Hence, there was significant motivation to study this largely unaddressed research area for invoking a thought process in the academic world in this direction.

1.3 **Research Aim and Objectives**

The aim of the PhD project is to improve the security of BI on Cloud Computing by developing a multi-party authentication framework for dynamic authentication interactions in a distributed environment, addressing the following research question – *"How to dynamically secure BI on the Cloud?"*

In order to achieve this aim, the objectives of the PhD project are defined as follow:

- *To investigate existing Business intelligence security solutions on the Cloud and identify the key security challenges for implementing BI on Cloud Computing*
- *To develop a multi-party authentication framework for dynamic authentication interactions between users and BI services in multiple Cloud systems located in different security realms*

- *To model the proposed framework and verify the correctness of proposed framework by using BAN logics*
- *To evaluate the performance of the proposed framework by using simulation and develop a prototype authentication system based on the proposed framework*

To achieve the above mentioned objectives, the following methodology has been employed:

Objective 1: In order to achieve objective 1: Background research is conducted in the form of critical literature review considering the security risks and challenges faced by cloud computing based Business Intelligence. This is intended to identify research gaps in this domain and provide a comparison between existing and proposed solutions.

Objective 2: In order to achieve objective 2: Use UML modelling language to focus on the design of a multi-party authentication framework, the development of five protocols and provide a formal specification of the authentication approach.

Objective 3: In order to achieve objective 3: Using well known BAN logic to formally analyse the correctness of the authentication security protocols, deriving the trust beliefs that form the principals of the protocol and to then verify authentication protocols.

Objective 4: In order to achieve objective 4: A set of experiments are implemented using the full production version of OPNET Modeller to evaluate the performance of the proposed framework. In addition, Eclipse is used as a contrasting method to develop the prototype authentication framework.

1.4 **Potential Contribution**

A multi-party authentication system is proposed. The proposed system addresses scenarios of Business Intelligence (BI) application access on cloud platforms. More

specifically, the proposal applies to the situation when members of different security realms need to access distributed BI services through a trusted principal. The authentication system is designed, implemented and tested using two high quality development tools, OPNET modeller and Eclipse. Using multi-party session management protocols, the authentication system allows two service instances to authenticate each other with their session memberships.

As a component part of the system, a session authority cloud (SAC) is proposed. The SAC is designed to control sessions in the cloud and precludes the concept of a home or foreign cloud, every participatory cloud obeys the decisions made by the SAC. The SAC comprises of an array of servers serving as a security vault. The vault holds authentication credentials and digital signatures for all of the tenants of all participating clouds. The root keys of the clouds are stored in a folder structure within the vault to identify the individual clouds. An active tenant will "know" the root key of its own cloud. The security realms (sub-domains) are distributed across all clouds and are identified through separate sub-domain keys. These keys are stored in subfolders within the corresponding cloud folders.

In the multiparty session scenario, members of multiple sub-domains may interact within a session. All such sessions will be identified by the SAC. The session keys will comprise of root key of the cloud, sub-domain key, and a section identifying the session. This means that there are multiple session keys which are valid for a given session; each session key has a common field for the session but unique fields for cloud root key and sub-domain keys. There is no need for any negotiation between the individual clouds because the SAC is aware of all participating clouds and their sub-domains.

The problems surrounding the implementation of multi-party session authentication to permit access to shared Business Intelligence data in a Cloud computing environment have been investigated. The issues include service instance identification as well as key generation

and distribution. Based on the investigation, a set of protocols for multi-party session management and cross-realm authentication have been proposed. Formal analysis of the protocols is carried out using the well-known BAN logic and a comprehensive empirical study is performed to assess the scalability and the runtime overhead of the authentication system.

1.5 **Organisation of the Thesis**

This thesis is organised into six chapters. This chapter (Chapter 1) presents the background and context, specific details about this research, aims, objectives, motivation, and significance of the study. Essentially, this chapter establishes the direction of this study.

Chapter 2 presents a literature review considering the security risks and challenges faced by cloud computing based Business Intelligence. This is intended to identify research gaps in this domain.

Chapter 3 presents a review on how BI and OLAP framework can be implemented on the clouds and presents key benefits of cloud computing for BI. Furthermore, it shows an approach for taking BI to the cloud as well as key challenges in hosting BI on cloud.

Chapter 4 describes details on simulation modelling tool and an introduction to the models studied in this research. Three simulation experiments using OPNET is described – cloud security, BI on the cloud, and BI security on the cloud. The chapter presents the results of simulations, comparisons and critical analysis of results. The chapter presents the results of simulations, comparisons and critical analysis of results.

Chapter 5 describes the design of the multiparty authentication system for BI, the mechanisms for hosting it on cloud computing, and presents the five authentication protocols in our multiparty authentication system. The correctness of the protocol is formally analysed and proven. The chapter presents the analytic assessment of the multi-party authentication

mechanism and illustrates the empirical evaluation of the multi-party authentication mechanism. A set of experiments are implemented on OPNET Modeller and Eclipse respectively, and the experimental results are used to evaluate the scalability, runtime overhead, and compatibility with the message level security protocols. Finally in this study, risks, ethics and legal implications are considered including mitigation factors and recommendations.

Chapter 6 provides the conclusion of the thesis and outlines future work. In addition, the limitations of this work.

1.6 **Main Publications**

- Al-Aqrabi, H., Liu, L., Hill, R., and Antonopoulos, N. (2014). "Cloud BI: Future of business intelligence in the Cloud", Journal of Computer System Science, Elsevier.
- Al-Aqrabi, H L Liu. IT Security and Governance Compliant Service Oriented Computing in Cloud Computing, Book Chapter: Principles, Methodologies, and Service-Oriented Approaches for Cloud Computing, in press, January, 2013, IGI Global, USA.
- Al-Aqrabi, H., Liu, L., Hill, R., and Antonopoulos, N. (2012). "Taking the Business Intelligence to the clouds". Proceedings of 14thIEEE International Symposium on (HPCC2012), Liverpool, UK, June 25-27, 2012.
- Al-Aqrabi, H et al. (2012). "Investigation of IT Security and Compliance Challenges in Security-as-a Service for Cloud Computing", Proceedings of 15thIEEE International Symposium on (ISORC2012), Shenzhen, China, April 11-13, 2012.
- .Al-Aqrabi, H et al. 2013. "Business Intelligence Security on the Cloud: Challenges, Solutions and Future Directions", Proceedings of 7th International Symposium on (SOSE2013), San Francisco Bay, USA, March 25 - 28, 2013.
- Al-Aqrabi, H., Liu, L., Hill, R., and Antonopoulos, N. (2014). "A Multi-layer Hierarchical Inter-Cloud Connectivity Model for Sequential Packet Inspection of Tenant Sessions Accessing BI as a Service" (HPCC2014). 20-22 Aug. 2014, Paris, France.

Chapter 2: Literature Review

2.1 **Multi party Authentication in Clouds**

Cloud computing offers shared applications/software, platforms, and infrastructure services on multitenant IT systems built on virtualisation. However, in the shared domains of Cloud computing, BI is exposed to security and privacy threats by virtue of exploits, eavesdropping, distributed attacks, malware attacks, and other known challenges to cloud computing (Al-Aqrabi et al. 2012). In a multi-tenancy environment, the authentication frameworks cannot be static. The business systems running on clouds are dynamic and hence the authentication interactions need to be dynamic, as well. Chen & Tu (2013) proposed a global authentication register comprising a privacy framework for tenants. The register can hold a private key and personalised data of each tenant for certification of a registration request. The system issues the private key after the personalised data has been provided to the registrar and matched successfully. There is a simple two-way interaction between the user and the cloud:

- Cloud asks for register key
- User notifies register key
- Cloud verifies register key
- On successful verification, cloud issues private key.

In cloud computing, the global authentication register may be viewed as the multitenant database holding primary records about the tenants registered and additional extended tables for recording unique detail about each tenant (Pippal et al. 2011). The primary table comprises details generated by the cloud service provider created for each tenant as per a standard format. The extended tables may record unique details provided by the tenants against a list of metadata classes (for example, names of spouse, dog, first school,

mother, favourite movie star, favourite colour, etc.). The extension tables help in validating private details about a tenant before issuing the private key for accessing his/her workspace. This concept is called "identity-based cryptography" (Li et al. 2009). The root key is the public key for unlocking a cloud or grid-based workspace. The root key is appended with the private key based on an identity-based signature generated by an interaction process comprising identity information entered by the client and the corresponding digital signature generated by the server. The signature is used by the authentication registry server to append private key fields in the root key and issue to the requesting client.

In cloud computing, identity-based cryptography and identity-based signature generation may be carried out by a separate array of servers (cloud array) dedicated for privacy-as-a-service (Li et al. 2011). The clouds may comprise hierarchical key structure as shown in the Figure below:

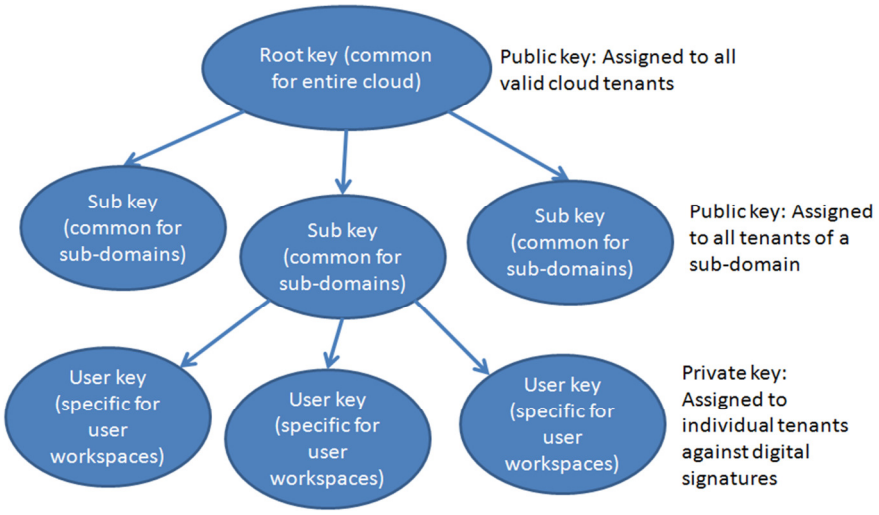

Figure 3: Hierarchical key structure in cloud computing

The key has two parts as public key fields and one part as private key field (Li et al. 2011). The first public key part is common for all valid cloud tenants, the second public key part is common for all tenants of a sub-domain within the main cloud, and the private key

part is for individual tenants that are determined as per individual digital signatures generated by a separate array of servers against private information provided by the tenants.

A cloud sub-domain is a group of multiple private virtual workspaces owned by an organisation or a group of related tenants (for example, a community or society) (Qin et al. 2013). A common public key for the sub-domain ensures that only the tenants owning its workspaces are allowed to access it (Qin et al. 2013). For example, the employees of an organisation will get access to the sub-domain public key by virtue of their employment records in the organisation. On the top of the sub-domain public key, a tenant-specific private key will be assigned based on private information provided by individual tenants. The below figure shows how this hierarchical scheme can ensure authentication of employees for uploading or downloading business data to and from the cloud, respectively. The private keys for employees may be hosted locally by the company that appends the private key portion with the sub-domain key generated by the cloud for the company. These two keys are finally appended to the root key of the cloud.

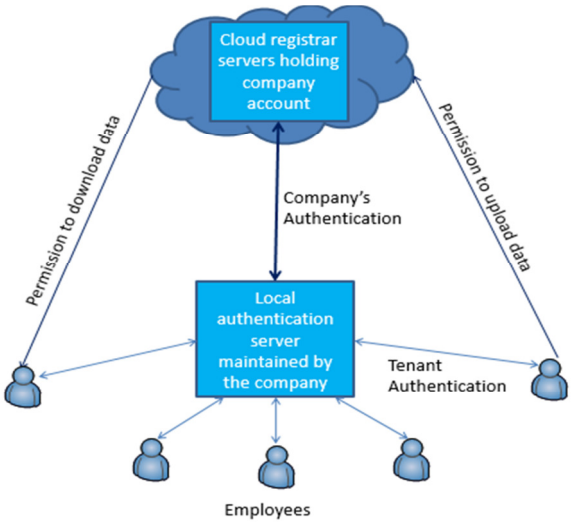

Figure 4: Hierarchical authentication structure in cloud computing

This framework may become complicated further when multiple parties coordinate within an authentication system for accessing resources stored on multiple clouds through the home cloud. In this scenario, a cross-cloud federation system may be established comprising discovery agents, match-making agents and authentication agents (Celesti et al. 2010). The discovery agent manages a process for discovering the resources requested by the parties by browsing the available foreign clouds. The match-making agent manages a process for short-listing foreign clouds to gain access to the resources requested. The authentication agent creates a security context and trust relationship between the parties and the home cloud, and between the home cloud and the foreign clouds, as shown in Figure 5:

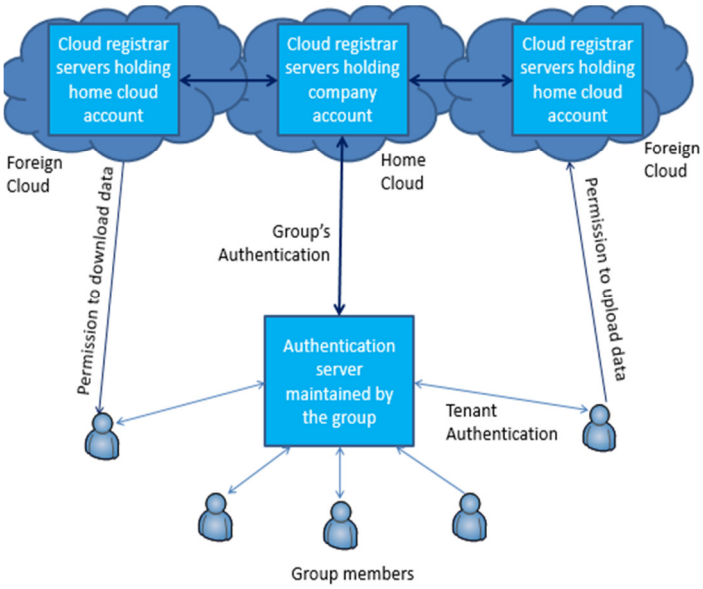

Figure 5: Hierarchical multi-party structure in Multi-cloud computing

There are multiple steps involved in this framework, stated as the following:

- The parties make authentication requests.
- The authentication agent sends queries for artefacts to all parties.
- The artefacts are resolved and a group-level public key is generated for all the parties.

- The discovery agent discovers the resources in the available foreign clouds.
- The match-making agent shortlists the preferred foreign clouds.
- The home cloud requests for public keys from the foreign clouds.

The foreign clouds make a request for artefacts before sending the keys, which the home cloud provides (establishing trust relationships between the home cloud and the foreign clouds).

On getting the keys, multiple keys are generated integrating the home cloud public key and the corresponding foreign cloud keys for accessing the resources.

The resources are collected by the home cloud and made accessible to the authentication server for the entire multi-party group.

The individuals in the party gain access to the resources through the authentication server by providing individual secrets and gain private keys. The private keys are appended with the group key that in turn is appended with the home cloud key.

As an alternative to the previous two steps, the home cloud can forward the groups artefacts to the foreign clouds such that they can provide individual public keys to the groups authentication server.

These group public keys assigned by different foreign clouds can be appended with the individual private keys such that they and be given access to the resources on an individual basis being a member of this multi-party group.

Such a group authentication protocol requires sharing of a number of security attributes, like secrets for group session key, secrets for private keys, secrets for key duplication (for resilience), secrets for session forwarding to other clouds, and secrets for control of keys (opening a vault of keys) (Dai et al. 2011). For facilitating automated inter-organisational processes, trust and dependability, and security needs to be ascertained (Avizienis et al. 2004). Dependability and security are interrelated through a number of

attributes (confidentiality, integrity, availability, reliability, maintainability, and safety) (Avizienis et al. 2004). There can be faults and errors in inter-organisational authentication processes leading to failure of services (Avizienis et al. 2004). There are higher chances of faults and errors in modern distributed business environments in which, the business specifications are dynamic and the runtime executing them are sometimes unpredictable (Xu et al. 2012). It is very difficult to standardise the runtimes as there may be "one-of-a-kind" execution of processes (Xu et al. 2012).

Xu et al. (2012) explained that participants in a service-oriented architecture (like Grid and cloud computing) may be parts of different security realms collaborating only at the runtimes. A Security realm may be viewed as a group of principals (like, people, services, and computers) engaged with an authentication system (a trusted principal) for accessing services and resources. It may not be possible that security realms participating in a collaborative transaction may have any authentication relationship directly. If authentication relationships are established among different security realms, the process may involve large numbers of extra and expensive steps for converting artefacts (like, validating multiple digital signatures). The federated authentication establishment may require time-consuming activities for negotiations and amendments. In many scenarios, this may be infeasible given that the participants in different security realms may be engaged in one-of-a-kind transactions only. A simpler mechanism may be to establish authentication paths through two distributed realms having pre-established trust relationships. The scenario shown in Figure 5 is an example of an authentication path. This process may overload the realms used for authentication path with conversion of artefacts. In addition, there are possibilities that the principals of two security realms trusting each other may not have authentication paths (enough trust relationships) for recommending certain principals. These possibilities may again require expensive negotiations and contractual amendments. In this context, Xu et al.

(2012) proposed an on-the-fly mechanism for establishing trust relationships for authenticating partners during runtimes. They employed the multiparty concept as described below:

In a multiparty concept, multiple parties can join or leave a session dynamically. The parties are allowed or removed from the session by a session authority. A simplified drawing showing the concept is presented in the below figure.

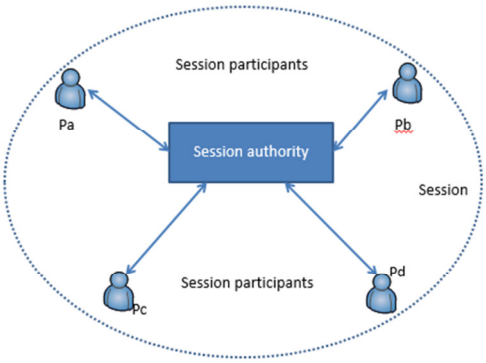

Figure 6: The multiparty session authentication concept

In this concept, a session authority controls the authentication of all session participants. Existing session participants can introduce new participants to the session authority for limited transactions. The session authority issues a secret session key to all running session participants. In practice the session authority communicates with multiple session handlers. Whenever a new participant joins or existing one leaves, the key is refreshed and shared with the active session participants using forward security techniques. The session authority recognises the session participants with the help of participant Ids. A participant leaving the session cannot reuse its participant ID for re-entering. Similarly, a reused ID will not be assigned to a new participant. The participants join through introduction only and need not share any secret artefacts to gain the session key. However, two participants acting as partners can share private keys using Diffie-Hellman algorithm.

In the framework of Xu et al. (2012), a cloud service provider can host an array of servers acting as session authorities. There may be multiple session handlers managing sessions among different groups of participants. The session key of a session may be a combination of three fields – root key (ascertaining cloud membership), sub-domain key (ascertaining membership of a valid security realm on the cloud) and the session key. This means that the sub-domain key portion will differ for each participant depends upon which sub-domain (security realm) it belongs to on the cloud. However, the session authority will recognise it as long as it is registered in the global registry of the cloud. Whenever a participant leaves, only the session key portion will be refreshed.

The process will become more complex when multiple clouds are involved. The process defined by Celesti et al. (2010) is quite complex given the number of exchanges of artefacts between the home cloud and foreign clouds and the home cloud and its tenants within the sub-domain. Their solution does not address the scenario when members of multiple sub-domains (principals of different security realms) want to interact (through a session) to access resources stored on multiple clouds. Hence, the framework proposed by Xu et al. (2012) needs to be applied to solve this problem. The roles of discovery agent and match-making agent are needed as-is. However the challenge is to ensure that the session authority is able to "convince" the foreign clouds without bulk exchange of artefacts. This can be fulfilled through a highest level of authentication relationship in which, the foreign clouds trust each and every tenant connected with the home cloud on the latter's recommendation. However this system puts a lot of ownership on the home cloud. Given that the resources are requested by principals of multiple security realms, the introductions made by session members become the primary approval points. Drilling down further, the final onus lies on the first introduction made by the session initiator given that all subsequent introductions made by additional members are trusted based on this initial introduction.

2.2 **Cloud Computing and its Security Challenges**

With the growing popularity of cloud computing, the concerns about security and compliance are also growing. Businesses do not want to be deprived of the already established and accepted benefits of cloud computing and hence they require continuous research towards the path to achieve standardised policies and controls on cloud computing that shall be acceptable to the regulatory bodies (Carroll, Merve and Kotze, 2011). It is important for the management of a business to understand what threats and risks exist on cloud computing infrastructures and what are the feasible mitigation strategies (Carroll, Merve and Kotze, 2011). In a survey conducted by Carroll, Merve and Kotze (2011), it was observed that the IT managers stated information security, business continuity and regulatory compliance as the top three concerns in moving their business workflows to the cloud. Ramgovind, Eloff and Smith (2010) argued that the full potential of cloud computing cannot be used for the benefit of businesses unless the security and compliance issues are sorted out. They further elaborated that secured connectivity to clouds over Internet, data segregation, data location and multi-tenancy are the key issues that are discussed by Gartner and IDC reports on cloud computing security that are coming in the way of achieving full compliance to the established regulations and acts. The main security issues to be solved in the context of connectivity, data segregation, data location and multi-tenancy are: identity management, authentication, authorisation, confidentiality, integrity, non-repudiation and availability (Ramgovind et al. 2010). At the technical level, Mukhin and Volokyata (2011) described that cloud computing comprises new types of vulnerabilities, like – incorrect provisioning in virtualisation, riding and hijacking of virtual sessions, insecure or obsolete cryptography keys, evasion of billing/metering data, data recovery of one user when the resource gets

allocated to another user, insufficient virtual network controls, poor authentication and authorisation in the virtual machines, etc.

2.3 **Cloud Computing Domains**

The architecture, deployment, workflows and service procedures of cloud computing is yet to be standardised. The academic scholars and professional architects have presented their own architectures of cloud computing in numerous research papers. (Qian, Luo, Du and Guo, 2009). NIST has come forward with a draft paper to standardise cloud computing, albeit currently at high level only. In the NIST's model, cloud has been presented as an integrated service oriented architecture comprising three forms of offerings – software-as-a-service (SaaS), platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS). NIST's model proposes that clouds can take four types of forms – public clouds, private clouds, hybrid clouds and community clouds. (Badger et al., 2011) The NIST's model is adapted and presented in the figure below:

Figure 7: The NIST model of cloud computing(NIST 2011) 'content removed for copyright reasons'

The key performance attributes of cloud computing comprises effective implementation of on demand self-services, broad network access, rapid elasticity, measured service and resource pooling (Badger et al. 2011). The key benefits from cloud computing hosting are: low-cost high-performance computing, usage based payments, improved performance of business processes, easier maintenance, world-class software tools at affordable costs, no hassles of upgrading, storage/computing on demand, better compatibility and portability of applications, better group collaboration and universal access (Miller, 2009).

Amburst et al. (2009) presented the following formula for calculating economic feasibility of moving business processes to the cloud:

$$\text{UserHours}_{\text{cloud}} \times (\text{revenue} - \text{Cost}_{\text{cloud}}) \geq \text{UserHours}_{\text{datacenter}} \times (\text{revenue} - \frac{\text{Cost}_{\text{datacenter}}}{\text{Utilization}})$$

This formula shows that the expected profit from the cloud should be greater than the existing profit from the self-hosted data centre. The formula reveals that if the utilisation of self-hosted data centre is higher, the feasibility to move to the cloud is lower. However cloud computing has some disadvantages as well. For example, it is not meant for businesses in remote locations that do not have reliable Internet connectivity (Miller, 2009). In addition, businesses under high security and compliance pressures should avoid cloud computing for time being (although they may find it feasible in due course) (Miller, 2009). According to Amburst et al. (2010), data lock-in, data confidentiality and auditing ability and unpredictability of performance are among the top five issues to be considered in the process of deciding on moving a business process and its data to the cloud. In this study, the author has focussed on these three issues of cloud hosting of business processes. In this context, the author has presented a review of existing solutions and also has recommended solutions based on self-interpretation of the challenges evident from the simulation exercise. Before getting into the details of these three issues, the author has presented a quick review of service provisioning on cloud computing.

2.3.1 Software as a Service

The SaaS framework on the clouds have been built using service oriented software tools that can run on any underlying platform and is mostly model driven. XML has been preferred for traditional computing devices and WML has been preferred for mobile computing devices for hosting application services. This is because the XML and WML formats are self-describable, easily discoverable, are not dependent upon a particular

programming language or platform, and can work with any database that supports them (Microsoft SQL, IBM DB2, Oracle, My SQL, etc.) (Sharma and Sood, 2011).

Bolze and Deelman (2011) presented the following characteristics of SaaS:

(a) The applications should be designed in such a way that they can effectively utilise large scale computing devices and storage.

(b) The applications should be highly sophisticated at the background but very simple at the user interfacing end.

(c) The applications and the underlying databases should support multi-tenancy (users and groups of large number of companies can work on the same application).

(d) The applications should release processing power and storage on demand rapidly in order to complete heavy duty tasks quickly.

(e) The applications should have provision for universal, domain specific and user specific master data tables.

(f) The applications and the underlying databases should support high levels of parallelism and parallel data access.

Ruiter and Warnier (2011) stated that multi-tenancy is viewed as the key security and compliance challenge on the cloud. As per them, many regulations require physical identification of data and its location in the IT systems. Some regulations stated by them are – Sarbanes Oxley Act, Gramm-Leach-Bliley Act, Health Insurance Portability and Accountability Act (HIPAA) and Payment Card Industry Data Security Standard (PCI DSS). The SaaS community needs to come forward with solutions to meet the compliance requirements directed by these acts. Aulbach et al. (2008) described that multi-tenancy can be implemented on the cloud by sharing the database schema among multiple tenants (user

companies), which is executed employing database object partitioning and employing a separate schema object to identify the objects owned by various tenants.

In the author's view, this is one of the effective ways of implementing data identification of multi-tenancy setups. The reports generated from the master schema objects can clearly identify which database objects belong to which tenants. In addition, to comply with the "location of data" requirements, the SaaS provider can ensure that the objects belonging to a company under compliance pressure can be distributed to only those servers that are physically located within the national boundaries. All databases have built in advanced features to control object distribution and their replications. A report of schema – tenant mapping combined with a report of physical location of objects and of the servers should satisfy the external auditors. Before we get into detailed discussion on such solutions, let us quickly review the application and infrastructure provisioning on the clouds.

2.3.2 Application Service Provisioning

Application service provisioning on a SaaS cloud is presented in the following figure, adapted based on an illustration by Mietzner and Leymann (2008):

Figure 8: Application service provisioning on cloud computing

(Mietzner and Leymann 2008) 'content removed for copyright reasons'

La and Kim (2009) discussed that service provisioning on SaaS clouds differs by users based on their workflows, privileges, logic. They further discussed that the resources pooled for service provisioning on SaaS clouds are hardware and networking, databases, web servers, platform services, specialised software servers (like use case testing tools) and application servers. The cloud resources are pooled and presented to user workflows through

composite resource provisioning carried out with the help of service provisioning engines. Some of the provisioning engines used by SaaS providers are Open QRM, Sun N1 provisioning manager and IBM Tivoli provisioning manager. The users connect to their business workflows through web services gateway, and the workflows in turn are served by the resource pool accessible through the service provisioning engine. (Mietzner and Leymann, 2008) The workflows are created employing standard tools like MPI, Apache Hadoop, Microsoft Dryad and Google Map-Reduce (Ekanayake et al. 2011).

2.3.3 Infrastructure Resource Provisioning

Infrastructure resources are provisioned in three ways – server virtualisation (Amazon), technique specific sandbox (Google) and a hybrid of both the techniques (Microsoft). Server virtualisation can be carried out using VMware, Xen virtualisation or Linux virtualisation solutions (Qian et al. 2009). However, many academic scholars view virtualisation as the key underlying technology for infrastructure resource provisioning on the cloud. Wan (2011) described that cloud based hypervisors should be deployed on multi-core CPUs on the bare-metal (the underlying real hardware) for optimisation of performance of parallel processing and queuing. In addition, Niyato (2011) presented that a virtual machine depository using multiple virtualisation solutions (OS layer virtualisation, Para-virtualisation, application virtualisation and bare metal virtualisation, as described by Wan (2011) can be the most effective solution to interface a private cloud with the public cloud (i.e., creating a hybrid cloud). In this context, Sotomayor, Montero, Llorente and Foster (2009) described that the host based virtualisation deployments need to be managed at the array level (an array comprises a large number of hardware devices) using software based virtual infrastructure management solutions. They demonstrated Open Nebula architecture that can be used for provisioning of hardware resources for the cloud users using sophisticated virtual machine

placement strategies with priority queuing based on immediate (prioritised) provisioning and best-effort (non-prioritised) provisioning (p. 19).

In a virtualisation setup, a server can host multiple virtual machines running their own operating systems. The operating systems of the virtual machines do not communicate directly with the assembly language of the hardware, but communicate with a middleware called “hypervisor”. The hypervisor interprets the instructions from multiple operating systems and translates them into consolidated assembly language instructions which are passed on to the CPU. Intel and AMD have developed special CPUs for running hypervisors for virtualisation ready servers (Phelps and Dawson, 2007).

2.3.4 Configurable Cloud Resource

The cloud users do not get opportunities to know about the pooled resources made available to them through the provisioning manager. They only have access to the user and development interfaces provided to them by the cloud service provider. The developers can use standard APIs to add functionalities to the workflows. The APIs do not facilitate development of an entire software product, but do allow the in-house developers of the user organisation to focus on the business logic and its processes, and create/modify the workflows and the interfaces (forms, reports and documents) (Chorafas, 2011). Examples of workflows that can be configured on the cloud are – user and privilege management workflows, service creation workflows (e.g. adding mailboxes), software development life cycle (SDLC) workflows, domain specific workflows (HR, Finance, Marketing, etc.) (Litoiu and Litoiu 2010). Litoiu and Litoiu (2010) further emphasised that the resource creation, modification and deletion on clouds should be managed through an in-house change management system in the user company. All users connect to service oriented virtual machines, and the services provisioning done by the cloud service provider (using tools like

Open Nebula or IBM Tivoli) on such virtual machines are accessed by the end users (Younge et al. 2010).

2.4 **Threats and Security Risks in Cloud Computing**

Cloud computing platforms may be subject to the following threats as described by Bisong and Rahman (2011):

(a) Unauthorised use of cloud computing components and resources

(b) Threats and vulnerabilities in the APIs provided to the developers

(c) Malicious insider trading

(d) All possible threats and vulnerabilities associated with shared IT systems and resources.

(e) Data manipulation, leakage and loss

(f) Hijacking of accounts or user sessions

(g) Other forms of unknown/emerging threats

The users view their virtual machine systems as self-sufficient desktops that are isolated from others. However, the virtual machines are hosted on servers on hypervisors. Hence, hypervisors are the targets of the attackers (Sabahi, 2011). It may be noted that hypervisors should be viewed as special purpose operating systems that are vulnerable to the traditional exploits that have been troubling operations systems for a long time, like – buffer overflow, DDOS, zero day attacks, viruses, spyware, covert channels, Trojans, etc. It is possible that an attacker can take a valid subscription on the cloud, take ownership of one or more virtual machines, and begin attacking other virtual machines. It may be noted that there are many exploits that need not be installed on the operating system, albeit they can just be launched from a folder. Hence, traditional security controls are needed on the cloud as well (Wen and Xiang, 2011; Sabahi, 2011).

2.4.1 Privacy on the Cloud

Following are the key privacy requirements on the cloud, as described by Katzan (2010):

(a) Identity: Identity on the cloud is an entity, which may be a virtual machine or a user or the objects that they are trying to access (an application, a data file, a folder, a document, or a record). For interaction of two entities, it is important that a trust is established between them. For example, a user is accessing a folder, a virtual machine connecting to a data file, a user trying to associate a record with a document, etc. In traditional self-hosted systems, there are multiple settings possible to establish trust between two identities such that they can interact or a session is denied (Katzan, 2010). The cloud systems should comprise the proven mechanisms of trust establishment between any two identities, like – identity management tools (LDAP, RADIUS, TACACS, etc.), RSA based public key cryptography issued and managed by a certification authority, secured socket layer, etc. (Ranganathan, 2010). NIST recommends Extensible Access Control Mark-up Language (XACML) and Security Assertion Mark-up Language (SAML) as the mechanisms for authentication and authorisation decision making between any two cooperating entities (Jansen and Grace 2011). Both SAML and XACML are emerging as large scale role mining and policy engineering standards for service oriented architecture, and hence are suitable for cloud computing (Takabi et al. 2010).

(b) Authentication: In every IT system, there are multiple levels of authentication depending upon the access rights of the user or the system. Clouds normally have distributed authentication mechanism to distribute load (Katzan, 2010). The SaaS provider will be accountable for all authentication levels till the users are allowed to the web apps servers hosting their workflows. The authentication and privileges

within the workflows and the backend database objects have to be managed by the user company. The SaaS provider will however be responsible to manage proven mechanisms of trust establishment between any two identities, whoever make the settings – users or cloud administrators. For example, if a workflow administrator makes privilege settings, the system supporting these settings is under SaaS provider's ownership. Similarly, if the users make privilege settings for access to database objects, the SaaS provider is responsible to make sure that they work. Overall, the SaaS provider is responsible for effective isolation of two entities, whether users, companies, virtual machines, group of objects in the database, etc. irrespective of who manages the security administration tools – a user or an administrator on the cloud. The roles should be very clearly bifurcated between the two parties such that during an incident analysis, it is clear who is accountable (Pearson 2009).

(c) Authorisation: Authorisation is the level of privileges assigned to a requesting entity, depending upon its role defined in the system. It is closely linked with authentication and the details are normally stored on the same systems used to manage authentication. Essentially, authentication and authorisation should be viewed as parts of the same security control because they cannot be delinked. On the cloud, the roles may be defined by the user company or the SaaS administrator depending upon the object. For example, the SaaS administrator will define roles in the web servers, whereas the user company should have an administrator defining roles in the workflows and backend database objects (Katzan 2010; Jansen and Grace 2011). NIST recommends that every user entity and virtual machine entity should have valid authentication and authorisation tagging. Guest accounts or stray accounts (not actively used by any entity) should be strictly prohibited (Jansen and Grace 2011).

(d) Accountability: This is the most complex challenge on cloud computing platforms. Given that the underlying systems are owned and managed by the cloud service providers, technically they are the ones responsible for any breach. However, there can be conflicts. For example, a user company may claim that the security settings in the workflow didn't work, or the cloud providers may claim that the user company didn't make the settings adequately in the workflow. Hence, essentially it is important that there is a high trust on the capabilities of the underlying platform. The security procedures of the hosting framework should be co-designed by the user company and the cloud provider (Pearson and Charlesworth 2009). NIST recognises this aspect and hence has tried to present a method. They have recommended that the cloud service provider should clearly mention the underlying platforms used for trust management, and the user company should clearly understand the capabilities of the platforms. They should accept the services only when they are satisfied with the platforms, or else use a third party for accessing the cloud (example, a unified threat management cloud provider). In general, each migration to the cloud should be treated as a change management project and an in-depth risk assessment should be carried out by the user companies. The cloud service providers should fully co-operate by sharing all the information about the platforms and technology (after filtering confidential information as applicable). The agreement between the cloud service provider and the user company should be based on the risk assessment and impact analysis. NIST recommends a number of best practices that the cloud provider should adopt (taken from traditional IT risk management practices) to make this process simpler. The accountabilities should be documented very clearly such that there are no conflicts during incident management (Jansen and Grace 2011).

2.4.2 Component Level Security on the Cloud

There are two key issues in component level security on the cloud:

(a) Trustworthy computing

(b) Auditing and compliance

Trustworthy computing should predictable, reliable and controllable and hence should be secured by design with high level of accuracy. At the component level, the trusted computing framework should have a trustworthy hardware, operating system, software tools, applications, maintainability and serviceability (Katzan 2010). As described by Shen and Tong (2010), the cloud computing components should benefit from the specifications developed by trusted computing group (TCG) in 2003, which comprises a number of hardware and software vendors. In this model, the hardware, operating system, software systems and applications are viewed as a stack of trustworthy systems certified by the TCG. For example, when Windows operating system is ported on IBM hardware, both vendors ensure complete compatibility and protection. Hence, it is essential that all the components deployed on the cloud should be the ones certified by TCG (Shen and Tong 2010).

However, it is important that the user companies should be confident about the cloud's trustworthiness. Hence, there should be established mechanisms for auditing the cloud's trustworthiness. Gul et al. (2011) discussed that the concept of third party auditor and security administration services should be implemented on the cloud. In this mode, these scholars have recommended different designs and topologies for connecting third party auditors or security administrators with the SaaS or IaaS providers' clouds. Carvalho (2011) recommended that identity management should be provided by separate cloud service providers called "security-as-a-service", using the traditional concept of "unified threat management" in service oriented mode. As explained by Chao et al. (2010), unified threat

management is a system in which a packet passing through is inspected against all possible threats before allowing to the destination host. However, the Gartner report recommends that virtualised data centres should have embedded security within the server arrays, by implementing security services on some of the virtualised servers and allowing the incoming user sessions to pass through them to the virtual machines hosting applications. This is because the inter-VM traffic within the same server (virtual networking) cannot be inspected by external network security devices (MacDonald, 2010).

In this study, the author has created an OPNET model comprising a separate security-as-a-service cloud (designated as UTM cloud) and tested the simulation of all user traffic passing through the UTM cloud to the application clouds. The results of simulation are discussed for analysing the feasibility of this solution vis-à-vis the solution proposed by Gartner report. The concept of security-as-a-service is good for transferring the risks to a service provider specialised on security only, rather than transferring the risks to a SaaS provider taking security as an embedded responsibility. This may ensure better compliance and better governance of security controls as per the risk assessment models of user companies. However, the practical feasibility also needs to be ascertained, which is the deliverable of the experiment conducted in this research.

2.4.3 Personnel Level Security on the Cloud

As per the above analysis, it is revealed that the personnel level security depends upon the personalisation of each user as an entity on the cloud. As discussed above, this depends upon the capability of segregating entities on the cloud, and maintenance of reliable trust management platforms. Trustworthy computing is the answer to this requirement, provided the components used for personalisation and segregation can be identified. In author's view, the cloud service providers should be able to produce tangible answers to a user's questions,

like – how do you ensure my login is protected, how is my login segregated from others, how is my data protected, how is my data segregated from others', etc. The third party auditor should be able to verify the personalisation and segregation effectiveness of the cloud service providers. Also, Unified threat management (security-as-a-service) may be an effective answer to personalisation and segregation.

2.5 **Security Solutions for Cloud Computing**

The steps recommended by NIST for security assessments before moving to the cloud are worth noting (Jansen and Grace, 2011):

(a) The user company should clearly identify the security and privacy requirements and define the criteria for selecting a SaaS provider.

(b) Based on the short listing of SaaS providers (and the related PaaS and IaaS providers), the user company should carry out detailed risk assessment using the information collected from the cloud providers and mapping with the internal control objectives.

(c) The capabilities and commitments of the chosen cloud provider should be very clearly recorded.

(d) All information assets being moved to the cloud should be listed. An acknowledgement against each asset should be taken from the cloud provider before moving them. The cloud provider should tangibly verify if the asset acknowledged by them has been moved to the cloud.

(e) It is advised that a competent legal advisor is involved when the terms of SLA are being drafted. Every line item should have clear accountability description, and no vague areas should be allowed. The jurisdiction of the conflicts should be clearly agreed, because clouds are global. All confidentiality and accountability clauses should be in line with legal and regulatory requirements of the jurisdiction.

(f) The agreement should clearly mention what and how reports will be published by the cloud provider and what and how the user company will audit at the cloud premises periodically.

(g) There should be detailed termination agreement. The roles of both the user company and the cloud provider should be clearly documented with the help of legal advisor. Issues like data lock-in and other possibilities of hindrances caused by the cloud provider in withdrawal by the user company should be clearly identified and addressed.

(h) There should be special clauses about how the cloud provider will return the assets to the user company (and vice versa) should be documented. Methods of data destruction and validation should be clearly stated. When the clauses are invoked, it is essential that both parties should follow all separation steps necessary under the regulatory framework to avoid any claims thereafter.

Based on these recommendations by NIST and the literature review in previous sections, let us analyse the three risk mitigation strategies, as recommended by a separate standard by NIST on risk management (Stoneburner et al. 2002).

2.5.1 Transferring the Risks

The process recommended by NIST is essentially a method of transferring risks to the cloud service provider. The user company may trust its due diligence done in selecting the cloud service provider and carrying out the risk assessment. In practice, the risk assessment will be based on information shared by the cloud service provider and its references (other customers). Hence, to a large extent the best that a user company can do is to ensure that the contract comprises all the terms from legal perspective, and the jurisdiction and contract enforcement aspects are carefully documented and agreed. There will be a significant element

of risk, which will be out of the control of the user company. Hence, the theories of trustworthy computing and effective trust management, personalisation and segregation practices will enable the user organisation to make a decision. The idea of security-as-a-service appears to be more promising because the risks will be transferred to a service provider specialised in security only, and it can be expected that the service provider has invested in state-of-the-art security products and solutions. However, the idea of all the user traffic passing through the security-as-a-service (UTM) cloud may not be feasible in practice. The author has tried to present this argument with the help of simulations carried out in thisstudy.

2.5.2 Absorbing the Risks

As discussed in the literature review, the systems will be owned by the cloud service provider. However, the security management practices will be shared between the cloud and the user company. For example, the SaaS provider or UTM service provider will take accountability of the authentication and authorisation services at the virtual machine level, but the user companies will have to take accountability of the security settings of the workflows. Assuming that the SaaS or UTM service provider has taken care of trustworthy computing effectively, all other risks pertaining to workflow security will be the accountability of the user company. Hence, a significant element of risks will have to be absorbed by the user companies as well. There should be effective security administration practices in the user companies as well, which should effectively interface with the service provider's security administration team (Al-Aqrabi et al. 2013).

2.5.3 Avoiding the Risks

As per NIST's recommendations, risks can be avoided by implementing effective security controls. In the recommendations pertaining to cloud privacy controls, NIST has

mentioned about internal control objectives of a user organisation. As recommended, these objectives should be effectively mapped with the capabilities of the cloud service providers before selecting the preferred one. If the cloud service provider is able to publish internal audit reports periodically that tangibly demonstrate the effectiveness of the controls, the user companies can treat the risks as avoided rather than transferred. This is because in a partnership mode, the internal auditors of the cloud service provider can be treated as internal auditors of the user organisations as well. Both the parties will be under compliance pressure within a jurisdiction and hence mutual partnership to avoid risks is a better arrangement than considering the risks to be transferred from one party to another. This arrangement will reduce the chances of conflicts and will increase the longevity of the service-oriented IT deployment for the businesses.

2.6 **Compliance and its Measurement**

Ruiter and Warnier (2011) stated that the only way to measure compliance is to allow third party auditing of cloud services by certified bodies or regulatory authorities. The cloud service providers will be under compliance pressure as much as the user organisations. Hence, they will definitely support third party auditing. However, as argued by Ruan et al. (2011), the third party auditors are currently not ready with tools, techniques and procedures to audit virtualised computing environments. In the current scenario, compliance can only be measured at the level of a user organisation, whereby the cloud provider will be required to demonstrate their capabilities related to trust management, authentication and authorisation, segregation, and personalisation. However, a complete auditing of the cloud providers will not be possible using traditional auditing tools used by third party auditors. Ruan et al. (2011) further stated that the auditors are not ready with tools for carrying out forensic analysis on

the cloud. Hence, the compliance measurement aspect of cloud computing is still inadequate and requires significant amount of development and testing.

2.7 **Summary**

The Cloud-based business processes contain collaborating BI services from multiple heterogeneous security realms which need to be engaged dynamically at runtime. If authentication relationships are established among different security realms, the process may involve large numbers of extra and expensive steps for converting artefacts. The federated authentication establishment may require time-consuming activities for negotiations and amendments. This framework may become complicated further when multiple parties coordinate within an authentication system for accessing resources stored on multiple clouds. The existing frameworks do not address the scenario when members of multiple sub-domains want to interact to access resources stored on multiple clouds. To address this research gap, this PhD project proposes a multi-party authentication framework for securing Business Intelligence on cloud computing. Specifically, the proposal applies to the situation when members of different security realms need to access distributed BI services through a trusted principal. Our proposed framework can authenticate dynamically and not requiring a lot of other processes for credential conversion that will need extensive invocations to intermediate services.

In this chapter, the security risks on cloud computing have been reviewed, and the unified threat management (UTM) and distributed models for securing cloud infrastructure are reviewed. In addition, the NIST recommendations on managing risks on cloud computing are reviewed and analysed in detail. The reviews in this chapter have resulted in a theoretical foundation about security risks and solutions in cloud computing. This foundation has been

useful in preparing the primarily OPNET models presented in chapter 4. The next chapter presents a detailed review of literatures on planning and hosting business intelligence on cloud computing.

Chapter 3: Business Intelligence on the Clouds

3.1 **Background**

Cloud computing is conceptualized in three forms – software-as-a-service (SaaS), platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS) (Convery 2010). The SaaS providers interface with the end users by virtue of provisioning of business application services similar to the ones that have been traditionally self-hosted by the corporate houses (Convery 2010). Cloud computing is getting popular because it offers many benefits over self-hosted IT infrastructures. The cloud computing users can discard the hassles of large scale investments in hardware and software platforms, in upgrading them regularly and in expensive licenses of application software used to run business processes, related transactions and decision-support systems (Brieter, 2010).

The end customers are required to pay for the services as per their usage of the underlying resources hosting the application and data services. This model has ensured better affordability of the best possible application systems thus supporting an increase in efficiency of businesses. The resources are allocated to end users against service requests made by their end terminals. The resources are allocated by a service provisioning engine that verifies the eligibility of the users from a separate schema object holding multi-tenancy data about all cloud users and groups. Once the eligibility is verified, the resources are reserved for the user through session bindings till the computing processes are in progress by the user terminal. The terminal is normally a virtualized client presented through a virtual server farm. However, there can be direct loading of resources as well (example, for data backup) (Bento and Bento 2011).

A separate layer monitors the session usage and utilization of resources such that the billing related information can be generated (Demchenko and Laat 2011). NIST is in the

process of developing standard protocols for user connectivity to the cloud through virtualization interface, terminal emulation interface, thin client interface and Internet browser interface. As of now, there is no standard protocol for users' connectivity to cloud hosted resources (NIST, 2011).

Business intelligence (BI) has been one of the most resource intensive applications historically. It comprises a number of data warehouses created by fetching decision-support data from organization wide databases. The data warehouses are updated at frequent intervals through appropriate queries executed on the business processing and transactional databases. An online analytical processing (OLAP) application fetches data from the data warehouses, organizes them in highly complex multidimensional data cubes, and presents to the users through user defined and configured GUI dashboards (Glaser and Stone, 2008).

BI and OLAP framework has a high business utility, because it helps in locating and eliminating/solving business process deficiencies, inefficient process steps and waste process steps. A BI and OLAP framework is expected to provide timely, accurate, organized and integrated information to business decision makers (Glaser and Stone 2008).

In spite of excellent business utility of BI and OLAP framework, many business owners were compelled to look for its alternative because of uncontrolled increase in computing and storage resource requirements in self-hosted environments. At some stage, the cost of maintaining and upgrading the BI and OLAP framework becomes unjustified for a business (Preston 2007). However, the unique selling points of cloud computing offers exactly what businesses need to successfully run BI and OLAP frameworks – unlimited resources, resource elasticity (resources on demand), moderate usage costs, high uptime and availability, high security, no hassles of upgrading and maintaining loads of servers and databases, etc. (Bento and Bento 2011).

Hence, it is hereby argued that cloud computing has the potential to offer a new lease of life to BI and OLAP framework. Moreover, it is also argued that cloud computing will also extend the power of BI and OLAP to small and medium scale businesses, which could not have afforded the framework in self-hosted IT infrastructures. However, it is important to establish a framework for implementing BI and OLAP on cloud computing platform. This research presents a literature review on how BI and OLAP framework can be implemented on the clouds and also presents its modelling and behaviour by virtue of an OPNET based simulation experiment. We have described how BI and OLAP framework can be modelled on a cloud and how it should behave in order to extend maximum utility to the businesses.

3.2 **A review of BI with OLAP on Cloud Computing**

3.2.1 BI and OLAP Framework

BI and OLAP framework comprises a highly complex multi-layer structure. Following are the key components of BI and OLAP framework (Glaser and Stone 2008):

a) A user interface layer comprising a large library of dashboards for graphical reporting.
b) A layer of data analytics comprising what-if scenarios, reports, stored queries and data models.
c) A layer storing the OLAP cubes formed by multi-dimensional data extraction from the data layer (the data warehouses).
d) A data integration layer for identification, cleaning, organizing and grouping of data extracted from the data warehouses before the cubes are formed.
e) A data layer comprising of the data warehouses.
f) A layer acquiring data from the business processing, decision support and transactional databases used by various functions of the organization.

g) The layer comprising the IT infrastructure components and related resources (data processing, storage and networking).

The key feature of a BI and OLAP framework is the OLAP cube, which is a multidimensional view formed in the structure of a matrix. The OLAP cube is a complex data view formed by running simultaneous queries on the tables of the underlying data warehouses that fetch at least three times more data compared with an ordinary database query. Each cube comprises a stack of multiple two dimensional reports (an ordinary planar graph showing a relationship between two variables). In typical OLAP applications, the queries fetch typically 10 to 12 times more data than an ordinary database query (Ross 2005).

An OLAP application may comprise multiple OLAP cubes stored in the form of a complex hierarchy of matrices having data organized in the form of cross-tabulations. The cubes are normally stored in separate data marts or within predefined tables in the data warehouses (Boutsinas 2005).

The common OLAP functions employed for formation of such cubes with a hierarchy of cross-tabulated data are drill-down, merge/split, roll-up, slice-and-dice and pivoting. Each matrix plane is identified by its own classification comprising different data mappings, like product codes mapped with product managers, product codes mapped with sales locations, locations mapped with revenues, revenues mapped with sales person, etc. The planes form a nest-like structure due to interrelationships. The resulting relationship looks like a tree with the roots comprising the primary variables and the branches comprising the secondary variables. For example, a product code is a primary variable and revenues generated in a sales location is a secondary variable. The dashboard operator can modify or change the primary and secondary variables, which direct the query to fetch different set of data to form different

cross-tabulations in the next querying cycle on the underlying data warehouses. Hence, the OLAP cubes are flexible and can be changed dynamically as per the business needs.

The following figure shows the BI and OLAP framework:

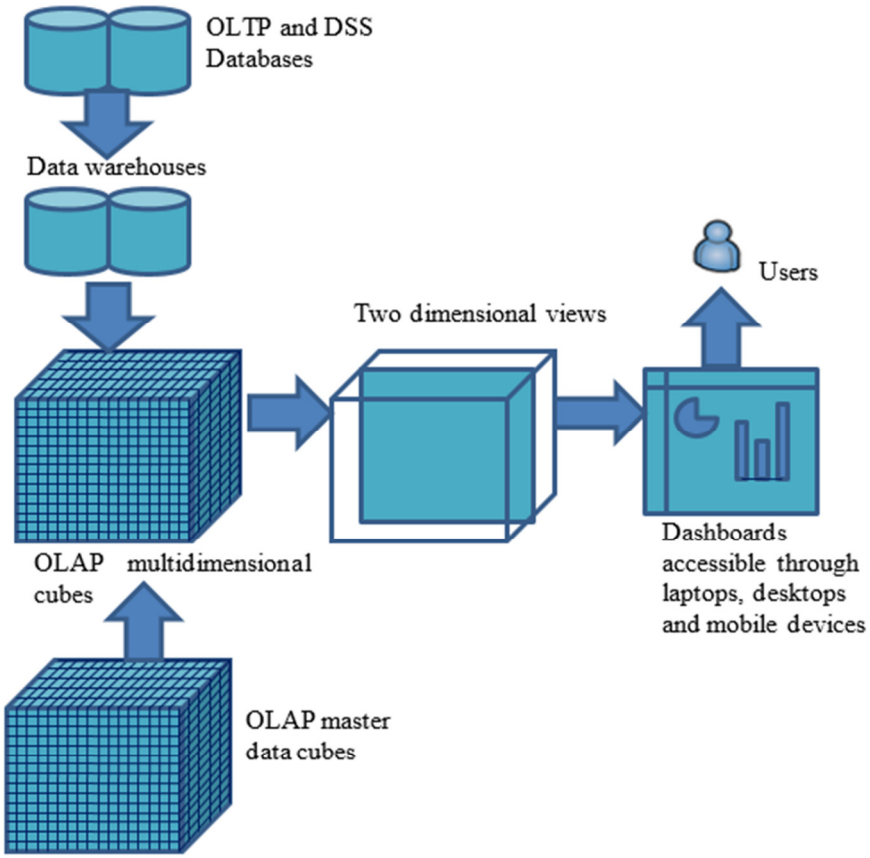

Figure 9: The BI and OLAP framework

Figure 9 illustrates two forms of cubes – the OLAP multidimensional data cubes and the OLAP master data cubes. The master data cubes control the relationship formation between the two-dimensional data planes within the multi-dimensional data cubes. A business user is offered a large range of variables that he/she can combine to form different views of two-dimensional reports needed in the dashboards. The data is pulled from OLTP (online transaction processing) and DSS (decision-support systems) databases into the data warehouse tables periodically, which in turn helps in periodic automatic updating of the data

in the data cubes and finally in the dashboards. Hence, the business users can closely monitor business performance by virtue of continuously updating dashboards. Appropriate colour coding of reference points/thresholds helps in generating alerts and alarms that helps the business strategic decision-makers to take appropriate steps (Cuzzocrea et al. 2007).

3.2.2 BI and OLAP on Cloud Computing

On the cloud, the matrices in the OLAP cubes can be formed using the web data warehousing concept making use of XML data files using DTD (document type definition) described XML programming language (Vrdoljak et al. 2003). The data structures in the cubes are formed using the DTD parsed XML files. The DTD format helps an XML file to exhibit relational properties of a conventional database. This is what enables the OLAP cubes stored on the cloud making use of XML data files following DTD structures (known as web cubes). This also helps the BI system to make use of web services components thus ensuring better performance on the cloud (Chadha and Iyer, 2010).

The entire OLAP framework comprising the dashboards and the data analytics layer can be hosted as SaaS. The BI and OLAP framework software platforms available for cloud hosting are SAP, IBM Cognos and Web-Sphere dashboards, Oracle business objects and Salesforce.com. The integration of data warehouses (XML based) and OLTP/DSS databases can be hosted on PaaS. The underlying servers, databases, storage and networking infrastructure components can be hosted on IaaS (Chadha and Iyer 2010).

The databases on the cloud need to be implemented in the form of a massively parallel system to support high demand elasticity of BI and OLAP framework. A centralized schema object may be designed to hold the details and privileges of all tenants on the cloud. The actual data files may be distributed among different schema objects. Each schema object holding the data files may be massively partitioned such that each partition can be held by a

separate server on a large scale server array. The IaaS provider should be capable of rapid expansion of the server array making use of virtualized array expansion. In this way, it may be possible to serve one partition through more than one server that can enhance the performance of BI. The IaaS provider should keep a close watch on the load distribution and response time patterns and make effective network changes to ensure that the network load is also distributed evenly (Curino et al. 2011).

The OLAP application hosted on the cloud may not be web services compatible. To make an OLAP application compatible to web services architecture, the SaaS provider may allow creating an intermediate layer to host a dependency graph that helps in dropping the attributes not needed in the finalized XML data cube (Aboulnaga et al., 2009).

Based on the reviews it is concluded that the key challenges in hosting BI on cloud are the following:

(a) Compliance of the BI application with web services architectural standards (and the standards defined by the SaaS or PaaS provider, like Google Apps standards)

(b) Deployment of massively parallel data-warehousing system with evenly distributed query load and even patterns of response times from all database servers. The IaaS provider should effectively use the virtualized server array management and expansion to meet the resources on demand.

(c) The network architecture should be designed in such a way that the query load can be evenly distributed among the servers in an array. This will ensure even query processing response times by the servers in an array. If the server array employs storage area networking for storing the XML data files and the OLAP cubes, the data fetching from various storage devices should again be evenly distributed by virtue of appropriate network connections.

BI hosting on cloud can ensure a distinct advantage pertaining to multidimensional faceted search on the underlying data warehouses. Every OLAP application supports multidimensional faceted search because of its ability to organize multidimensional information in the form of cubes. A faceted search helps in reducing information overload because it gives the user an opportunity to choose multiple category filters for searching a particular information using drill down feature of OLAP search function. For example, if a user wants to buy a new car, he/she can define multiple facets (like, make, model, year, city, and colour) to fetch multidimensional information from the data warehouses through organized filtering. This method helps the user to make use of wide scale of information from the data warehouses through a structured query (Kashyap et al. 2010).

Faceted search is highly resource intensive, especially when a large number of facets are defined in a structured query. In the past, researchers have been struggling to optimize the performance of faceted search tools sitting on the top of OLAP cubes in self hosted data warehouses. However, the service oriented architecture on cloud computing is an ideal platform to accelerate the performance of faceted search on OLAP cubes by virtue of massively parallel computing resources. The cubes in services oriented architecture is formed in the form of multidimensional XML data files, as discussed in the beginning of this section. The faceted search can be created in the form of a hierarchy of facets. The search results may be in the form of a set of multidimensional results with zoom-in and zoom-out options. These results may be stored in temporary view tables. On the cloud, the OLAP cubing operations can be carried out on the search results to extend the visibility of faceted search to the dashboards. These cubes may be temporary and serve the purpose of temporary drill-down or slicing/dicing until the user is successful in deriving the desired results. The mechanism is highly resource intensive and hence the massively parallel processing ability of cloud computing can make its usage feasible (Lempel and Sheinwald 2010).

3.2.3 Benefits of Cloud BI

Nowadays, Cloud BI solutions are gradually gaining popularity among businesses, as many businesses are realising the benefits of data analytics. Businesses need quality insights driven by accurate data more than ever. The SaaS providers are serving as the primary interfacing to the business user's community. Cloud BI is the concept of delivering BI capabilities as a service (Al-Aqrabi et al. 2014). The following are key benefits of Cloud computing for business intelligence.

3.2.3.1 Cost efficiency

In the Cloud, companies do not need to budget for large, up-front purchases of software packages or carry out time-consuming updates on local servers to put the BI infrastructure up and running. They will treat it as a service, paying only for the computing resources they need and avoid costly asset acquisition and maintenance thereby reducing the entry threshold barrier.

3.2.3.2 Flexibility and Scalability

Cloud BI solutions allow for greater flexibility to be altered quickly giving technical user access to new data sources, experimenting with analytical models. With the Cloud BI solutions, business users will be able to keep a better fiscal control over IT projects and have the flexibility to scale up or down usage as needs change. Moreover, in the Cloud, resources can automatically and rapidly scale in and scale out, and it can support large numbers of simultaneous users. This means that customers can easily increase their software usage without delay or the cost of having to deploy and install additional hardware and software.

3.2.3.3 Reliability

Reliability improves through the use of multiple redundant sites, which can provide reliability and secure locations for data storage and the resources can be spread across a large

number of users, which makes Cloud computing suitable for disaster recovery and business continuity.

3.2.3.4 Enhanced data sharing capabilities

Cloud applications allow data access to be shared remotely and enable easy cross-location data sharing capabilities as they are deployed via the Internet and outside a company's firewall.

3.2.3.5 No capital expenditure

Low TCO (total cost of ownership) is a key benefit of the Cloud model. With the Cloud, companies pay for a service they actually use. With this policy, Cloud computing allows companies better control the CAPEX (capital expenditure) and the OPEX (operations expenditure) associated with non-core activities. Hence, the benefits of BI can be rolled out faster to more users within the organisation.

3.3 **Business Intelligence Security on the Cloud**

Business intelligence (BI) is a complex framework in which, relevant data units are captured from all the transactional and decision-support databases in separate data-marts and data warehouses. The key components in a large-scale enterprise-class BI framework are the following (Lehner 2007):

(a) Intelligent data extraction agents are installed in all the transaction processing databases and decision support systems of the organization. These data extraction agents follow certain rules defined by the BI strategist in a centralized data extraction engine controlling the agents. The process of data extraction may comprise of a selective export of database objects, or an SQL program with

appropriately defined select statements. Schiefer, List, and Bruckner (2002) emphasized that even external data sources may be needed by the business owners. The data from external sources may be collected in manual formats (like Excel) or sent by consultants in pre-defined formats.

(b) The extracted data is loaded into temporary data marts. These data marts are accessed by expert data modellers to ensure that the data is transformed as per the standards set for the data warehouse. Golfarelli (1998) explained that a number of design components are needed in a data warehouse design, like – facts, dimensions, hierarchies, attribute tree, pruning rules, grafting rules, temporal and spatial designs, and OLAP cube structures.

(c) The data needs cleaning and transformation into the desired formats before it is loaded into the data warehouse. Schiefer et al. (2002) emphasized that there should be a change control process to ensure that the transformed data is approved before loading into the data warehouse.

(d) After appropriate cleaning and transforming, the data is finally loaded into the data warehouses. SQL loader and object import are used at the core level. However, they are ably controlled by advanced data loading tools eliminating chances of errors.

(e) The data units are accessed from the data warehouses through multi-dimensional OLAP queries. A number of interfaces are in action. Examples are software, hardware and database interfaces, front-end interfaces (dashboards and special purpose report customization and presentation screens, such as crystal reports), and data visualization (export to excel, adobe acrobat or other formats).

From this analysis, it is clear that business intelligence is not just an IT enabled system but comprises a number of processes requiring human interventions and interfacing.

3.3.1 BI Security on the Cloud

Malinowski and Zimanyi (2008) explained that the rules of intervention (for data extraction, transformation and modelling) change periodically as per the business rules and the expectations of the business users. Hence, a lot of activity is carried out in changing the rules of data acquisition, data transformation, formatting, and finally loading. Hence, security challenges in BI is not just limited to technical settings but also related to procedural and human security. In this study, the focus is on technical security challenges and solutions when BI is hosted on cloud computing. Hence, the review from this point onwards is focused on technical security of BI. In the end, two OPNET models have been presented. The first model demonstrates security controls for BI on the cloud employing a separate cloud delivering security-as-a-service using unified threat management. In the second model, a distributed security controls framework is presented in which, the technical security controls are closely integrated in the database server arrays and the application server arrays. A comparison has been presented using custom reports in OPNET to evaluate which option is suitable for BI security on the cloud.

3.3.2 BI Security Challenges and Controls

A security architect for BI may be challenged to implement all possible technical controls at various stages of the BI process. A BI framework will need access controls at the hardware systems, at network systems, at the instances and objects of data marts and data warehouses, at the metadata repositories, at OLAP servers, at the data view systems, at the data presentation layer, at the application services layer, and all layers of authentication (Kadan, 2012).

Farhan et al. (2012) emphasized that security rules are needed within the objects and closely tied with the tables holding temporary as well as permanent data. These rule tables are

needed in data marts employed at the extract level, the data marts at the data transformation and cleaning level, and the data warehouses at the loading level. The rules comprise of permissions for the human agents involved in extraction, transformation and loading. The rules also comprise read and write permissions on the temporal and non-temporal data. As an example, Farhan, Marie et al. (2012) demonstrated how security constraints could be applied at object level to control and log formation of dimensions.

Fernandez-Medinaa et al. (2007) presented a similar scheme for securing data warehouse objects. For running multi-dimensional OLAP queries, the objects in the data warehouse may be implemented in a multi-dimensional model. Fernandez-Medinaa, et al. (2007) proposed that security controls need to be integrated with the conceptual multi-dimensional modelling by including objects describing and enforcing security constraints on various operations requested on the objects. They proposed a system of secure UML with extended modelling using object security constraint language (OSCL). Given the multi-dimensional proposition of implementing security constraints, it is possible that security objects will also attain a hierarchical shape for securing many-to-many relationships within the hierarchy of various dimensions. Hence, security constraints need to be extended to making and breaking relationships in addition to transformation requests and data access requests. They illustrated this concept through a hierarchical structure of a multidimensional model in which, field level restrictions have been implemented through the multi-dimensional access control rules.

The tight object level security controls presented by references Farhan et al. (2012) justified that given that the data has been extracted from the domain owners through the extraction agents and until the time it is loaded into the data warehouses, the temporary databases do not have owners of different data units (Brankovic et al. 2000). In practice, BI users are secondary users of the data units and hence may not be able to own their security

because the data units are flowing in continuously. Thus, there may be a threat of lack of information on handling a data unit (Brankovic et al. 2000). This mandates that the controls on the data units in temporal and non-temporal databases should be more stringent that the original data stores from where the data units have been extracted. However, such tightly coupled security in the multidimensional objects and the query, transformation, loading, and views processes makes the overall BI very heavily loaded with security-linked objects. There may be further rules engine for securing OLAP cubes positioned above the data warehouses (Blanco et al. 2009). Ahmad (2010) recommended that data might be encrypted during the transformation process before loading into the data warehouse.

The above analysis presents a challenge in implementing reliable and effective security controls in BI. It may appear that a combined security framework comprising multidimensional security architecture within the database objects, hierarchical controls on all relationships and data encryption at the object level may produce a significant volume of security related objects in the BI framework. This may make the BI framework heavier on computing resources and capacity hungry. Moreover, the BI framework may significantly overload the network as well due to added security validation overloads. It should be kept in mind that even the network will comprise of its own security-linked overheads (for example encryption on the LAN, MAN and WAN links).

Stobla et al. (2005) has presented a solution to this problem in the form of a cluster of federated data warehouses connecting to separate network switches and storage hardware. Such a system may comprise multiple data warehouse implementations in multiple servers through a process of depersonalization. The framework proposed by Stobla et al. (2005) is redrawn and presented in Figure 10 below:

Figure 10: A federated data-warehouse system to facilitate distributed security (Stobla et al 2005) 'content removed for copyright reasons'

The framework proposes creating a distributed data warehouse system comprising federated database servers owned by different administrative groups. The federated systems ensure that critical data is striped into multiple data elements whereby, an individual data element does not make sense to any user unless all corresponding data elements are combined. This concept of striping critical data is termed depersonalization. The sensitive data units or their groups are first identified and then striped smartly to ensure that one or more elements do not reveal the information hidden in the data units after they are assembled. In the second phase, called pseudonymization, all re-identification requirements of a data unit are eliminated. For example, if the data unit comprises a social security number matched with the age, first name and last name of an individual, then after splitting this information, the programmer need to ensure that they need not be integrated at an intermediate stage before reaching the authorized user. The third step is federation, in this phase the OLAP queries are federated in such a way that it fetches data stripes from the databases and assembles them only at an authorized user's workstation. This system gives an impression that the authorized users are accessing a virtual data warehouse which is invisible to unauthorized personnel (Stobla et al. 2005).

A closer analysis of this model reveals that federating the data warehouses and stripping the critical data into multiple data elements fetched and assembled through a federated OLAP query may return similar results compared with a virtualization system. Hence, keeping in view the recommendations by scholars about security solutions it appears that BI on cloud can address the security needs. This is because BI itself is very resource hungry and its security overheads are much heavier than other relational database

applications. Abadi (2009) argued that relational databases could be made highly elastic on the clouds, and hence all these additional security related schema objects can be easily implemented and expanded on the cloud. In self-hosted mode, such schemas may simply add to the overheads, increase cost, and cause reduction in efficiency and effectiveness of the entire system.

Before undertaking a review of BI on the cloud, a review of OLAP security is presented herewith. OLAP is a dynamic multidimensional reporting application. The data is first consolidated in the form of views comprising data cubes, and then presented to a dashboard application that can facilitate multiple presentations of data for the end users. A basic OLAP view comprises of a services cube and a user cube. The security settings in an OLAP system may be based on access restrictions to whole cubes or parts of a cube (Priebe and Pernul, 2000).

OLAP security controls may also comprise permissions to carry out OLAP operations, like – roll-up, drill-down, drill across, set operations, selection, change base, projection, and slice-and-dice. This can be achieved by using multidimensional security constraint language (MDSCL) incorporating creating cube, hiding cube, hiding measure, hiding slice, hiding level, and hiding measure where an attribute is present syntaxes (Priebe and Pernul, 2001). The distributed security controls model is redrawn and presented in Figure 11 below:

Figure 11: Distributed security controls in an OLAP cube comprising viewing
(Priebe and Pernul 2001) 'content removed for copyright reasons'

Figure 11 presents basic and advanced controls those can be applied to an OLAP cube. These controls can be pulled from a centralized logical controls engine and applied to various cubes created by the users. A data security administrator or even the users can pull and apply access controls. The controls engine needs to be designed at the logical layer of the OLAP and relational database systems, and the pull-down menus needs to be designed at the conceptual level where the multidimensional data modelling has been carried out (Priebe and Pernul 2001).

Based on these fundamentals, the OLAP system can be made to preserve privacy of users based on a new protocol called Secure Distributed-OLAP aggregation protocol (SDO). Such an OLAP system can generate privacy-preserving views in addition to local data views. Such views present the identity of the owners those have established or manipulated multidimensional relations (a form of object and view-level activity logging for manipulation of multidimensional relationships) (Cuzzocrea et al. 2012).

The advantage of having such a view is that any unauthorized manipulation of structures and dimensions (apart from the team engaged in transformation tasks) can be detected. The privacy view will present details about the “privacy metrics” captured from the user attributes against each committed operation on the database objects. It will be in the form of a tree that follows the hierarchy structure and related dimensions built by a transformer (Agrawal et al., 2005).

Wang et al. (2004) explained that the privacy metrics could be termed as inferences that analyses the source of sensitive information and the target where it is requested based on a sensitivity criterion and logs the personal details in the transaction log. This phenomenon will be based on the original privacy settings of the source transactional database (Wang 2004).

Fienberg (2006) proposed more controls built over privacy-preserving features built in the data marts, data warehouses and OLAP cubes. The recommendations include deployment of intrusion detection and prevention systems (IDPS) deployed close to the databases, authentication and authorization repositories (like LDAP), and the temporary stores for OLAP cubes. In addition, Fienberg (2006) recommended deploying secured socket layers (SSL) based user access systems, and encryption of critical data elements including objects holding privacy preserving information. The idea is to implement selective revelation through security barriers based on a calculation of risk-utility trade-off and need-to-know-basis strategy.

The above reviews indicate that scholars have preferred distributed and embedded security components in BI components rather than centralized security controls (like the unified threat management framework). However, the studies have not presented evidences on how the distributed and embedded security components are expected to perform in comparison with a unified threat management approach, given that the computing and networking resources are always limited. In this study, the two approaches have been tested by modelling them separately in OPNET and comparing the simulation results. Given that this is a study about BI security on the cloud, the modelling is done to reflect the cloud based database and OLAP server arrays. In the UTM approach, a centralized database monitoring system has been implemented in a security-as-a-service model. This system owns a system-level root login into the database instances holding the objects and logs all DDL and DML commits along with the identity of users finalizing the modifications. Such a database security monitor is very common in legacy systems (like Oracle Enterprise manager). A separate database of activity logs is maintained by the security administrators. Exception reports are generated periodically by the security administrator to locate the transactions committed by individuals not supposed to have access to the objects. In the second model,

there is no UTM cloud, and all security components are embedded within the database and OLAP application servers.

To build the models, a theoretical fundamental understanding about BI on the cloud and security issues in cloud computing is needed. Hence, the next two chapters are dedicated to establish the theoretical foundation needed on a cloud.

3.3.3 Securing BI on the Cloud

Any BI expert will visualize that moving BI to the cloud will involve a massive implementation of relational databases on the cloud hosted servers. Aboulnaga et al. (2009) presented a number of design considerations in deploying database appliances on the cloud. First, it needs to be considered how databases are deployed in a self-hosted or ISP hosted environments. A number of servers are normally deployed to facilitate load sharing and fail-over. However, the database instances are tied to single servers, or at the most clusters of servers. The following figure demonstrates how databases are implemented in self-hosted IT infrastructures (Aboulnaga et al. 2009):

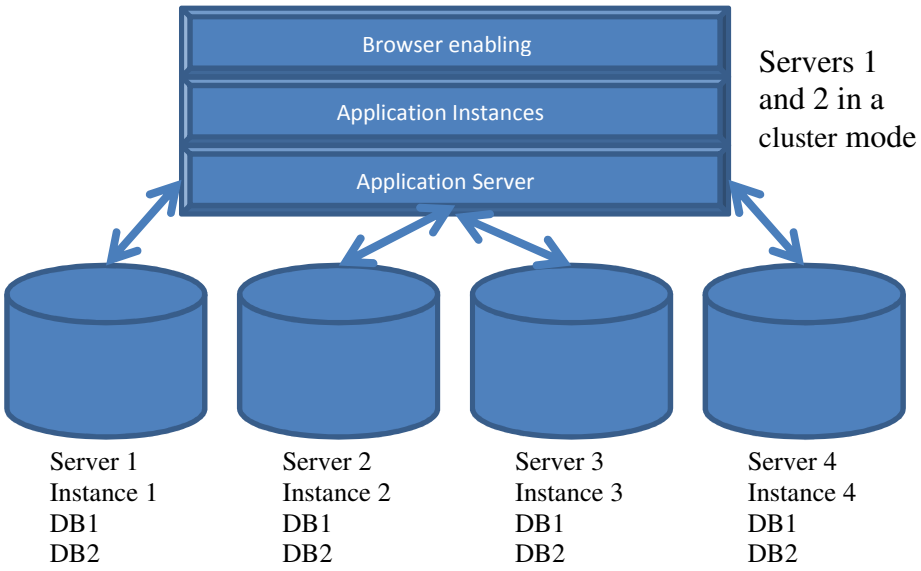

Figure 12: Database appliances in self-hosted environment

In a self-hosted setting, each server holds a separate instance and copies of all databases is replicated within them. The advanced features of the database tool are used to create a clustering among all instances, each holding the same set of objects. The databases are kept synchronized with the help of advanced replication features. Given that, all the databases have a common set of transaction logs (due to advanced replication), the failover is implemented by applying the latest transaction logs on the remaining servers when one or more servers fail (Aboulnaga et al. 2009).

This scenario is completely changed when the database appliances are hosted on a cloud. A presentation of relational databases on cloud is shown in the figure below:

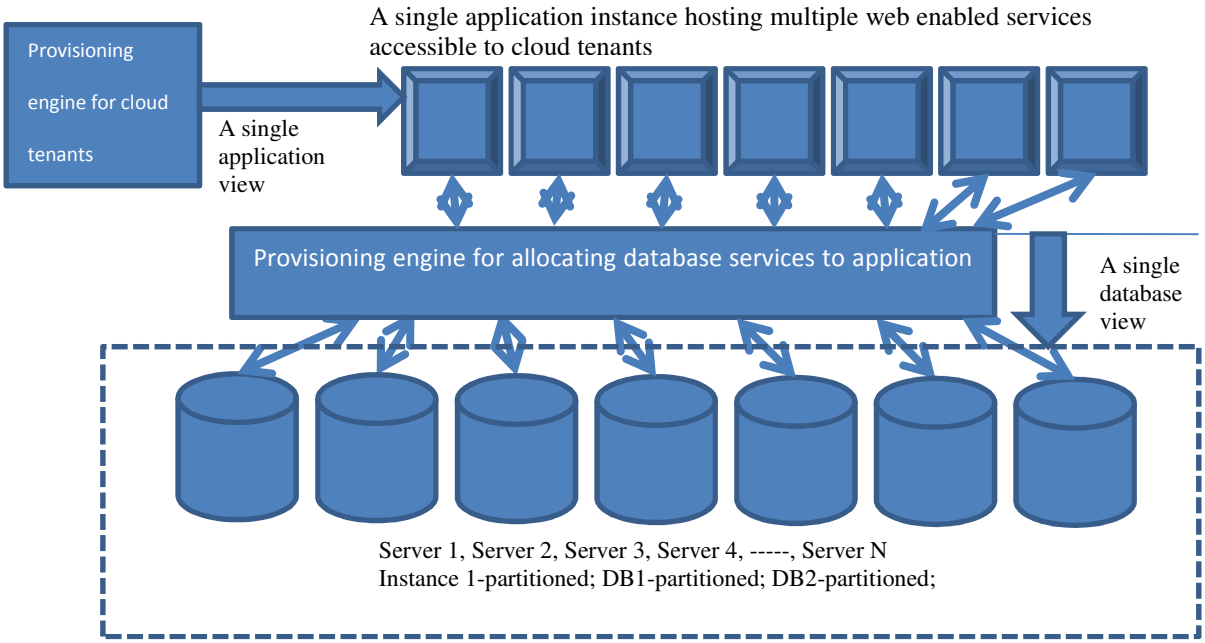

Figure 13: Database appliances in cloud environment

On a cloud, the database servers are connected in a large and flexible array. A single instance, and all the databases and their objects held by the instance, is partitioned to ensure that each server services one or more partition (Abadi 2009). The server arrays (both application and database) are implemented in the form of a massively parallel processing

system with no physical object tied to any specific hardware. This means that in a database array, each server will hold a partition of all data files in the instance (Al-Aqrabi et al. 2012).

The data warehouses or temporary data marts on the clouds need not serve the applications through data files in their legacy formats. The applications are hosted in a web enabled services oriented architecture. Hence, the data files presented to the applications may be in XML format. Hence, all data warehouses on the cloud may employ XML files comprising the intended hierarchies and dimensions (Vrdoljak et al. 2003).

It is possible to use XML files as data cubes because of their hierarchical structure and ability to hold multidimensional views (Hummer et al. 2003). The database resources are assigned to the application servers through a provisioning engine facing the array of application servers, and the application services are provided to cloud tenants through another web-services provisioning engine facing the tenant machines. This model is called database-as-a-service on the cloud (Curino et al. 2011).

The temporal and non-temporal data files in data mines as well as data warehouses are stored in the form of relational XML files. The OLAP views are generated by creating XML hierarchies as per the object requests chosen in the application. The requests are embedded into a XML query and the corresponding XML file is generated by the data mart or warehouse (Wang et al. 2010).

3.4 **Summary**

In this chapter, a review of BI architecture and mechanisms for hosting it on cloud computing is discussed. The knowledge gained from this review helps in understanding how BI can be hosted on cloud computing and how its architecture is different from traditional LAN hosted BI architectures. In the next chapter, a review of BI security risks and solutions is presented.

The reviews in this chapter have created a good insight about security threats in BI and the strategies for addressing them. In addition, the reviews have given a good overview about BI security controls in a cloud computing environment. These reviews have been used as inputs for creating the third and fourth models in OPNET. In the next chapter, a review about OPNET modelling and details of the models pertaining to this research are presented.

Chapter 4: Primarily: Modelling and Scenarios

4.1 **Introduction**

This chapter is divided into two parts. In the first part, an overview of OPNET tool and review of simulation project using this tool is presented. In the second part, four simulation experiments using OPNET is described – cloud security, BI on the cloud, BI security on the cloud and multiparty authentication system for securing BI on the cloud. These experiments present a simulation based perspective of applying theoretical knowledge gained from literature review in practical scenarios such that the factor contributing to an optimum design for securing BI on the clouds can be derived. The results of these experiments have been discussed in this chapter.

4.2 **Research Methods**

Research methods are vital for the success of a project. This section describes the study modelling tools and Integrated Development Environment (IDE) that used within the project. The authentication system is designed, implemented and tested using high quality development tools, OPNET academic version, OPNET Modeller 14.5 and Eclipse. In this study, a set of experiments are implemented respectively to test and assess different scenarios of BI on the cloud that addresses the objectives.

Research methods are vital for the success of a project. The possible research methods include empirical work, case studies analysis, questionnaire, and simulation

Empirical methods are mainly based on empirical evidence. It is about gaining knowledge by means of direct and indirect experience or observation. Empirical studies; quantitative and qualitative approaches are part of the critical paradigms in empirical research. First, in qualitative research, the researcher is mainly concerned with studying

natural settings of objects. It is vital to note that in a qualitative method, the researcher indulges in interpreting phenomena using descriptions derived from other persons. Among the internal traits of qualitative research is the fact that there are diverse ways of interpretation of the occurrences under investigation. Thence, qualitative methods associated with determining causes identified by participants the research while aligning with their ideas regarding the underlying problem matter (Wohlin, 2012).

On the other hand, quantitative research deals with quantifying associations or determining uniformity among groups with the purpose of obtaining an action and reaction trend among the groups of participants. As a means to effect quantitative research, controlled experiments are set up for the collection of the required data for the research study. Notably, therefore, the use of quantitative methods hence relies on measurement as an underlying constant in research (Conradi, Wang & Esernet, 2003). Therefore, quantitative inquiries serve as the vital tool in determining the effect of alterations on the variables under study. As a research tool, quantitative data is advantageous as compared to other methods given capability to permit the researchers to examine and analyse the data using statistical tools.

Case studies analysis is critical in the collection of previously unobtainable data by use of typical methods in research. Through case studies, data collected is of upscale quality and more reliable. Due to the in depth and reliability of the data, the method is thus more preferred in comparison with other investigational designs especially when researching on large samples lacking similar participants. However, although the method has the merits, the fact that data collected generalisation for a wider population under study is impossible makes the procedure of case studies unattractive to some researchers. The disadvantage thus makes data e collected over longitudinal case studies irrelevant this not viable in the study. Despite the generalisation loophole, other limits of cases studies are the fact that they only describe behaviours detected without an explanation of how the acts occur and manifest themselves.

Also, drawing definite cause/effect seems impossible in the event of using case studies as a method (Conradi, Wang & Esernet, 2003).

Another useful investigative method involves questionnaires. For researchers interested in vast amounts of information, questionnaires are a critical resource especially when there are time restrictions thence vital in collecting data over a short span of time during a survey at a manageable expense. Likewise, the outcomes of questionnaires are usually fast and easy to quantify by use of the various statistical packages and analytics tools by the researcher. However, given the open-ended structure of the questions contained, the participants are thus bound to read and interpret the various included inquiries in a different way thence leading to different responses.

There exists other concerns during research such as the potential of the members to forget some of the vital parts to be included in the replies and hence not fully answer the questions satisfactorily. From the responses, researchers need to determine the vitality and relevancy of the answers which at mostly becomes time-consuming and causing a delay to primary study (Wohlin, 2012). Similarly, the determination of the genuineness and truthfulness of the responses by participants is difficult as the opinions are subjective across the different interviewees.

Simulation has been selected in this PhD project because all above methods cannot be used to evaluate the performance and overhead of the proposed systems.

Simulation is used almost exclusively in this type of research as the data and processes under consideration are volatile, high value, commercially sensitive and therefore datasets are not readily available for public use. To implement and test a security framework in real commercial cloud systems is prohibitively expensive which cannot be affordable by this Ph.D. project. Therefore, simulation is a suitable research method for this PhD project.

Eclipse is an integrated development environment (IDE) for Java and other programming languages like C, C++, Python, and PHP etc. Then the Java source code will be imported to Eclipse for editing and debugging. Eclipse is produced by Sun Foundation as a free, open source tool which is readily available for Eclipse users (Draxler et al. 2015). It has been used to develop authentication model in chapter 5.

OPNET academic version was used as part of the initial research for the primarily modelling scenarios in this chapter. This version is free but has scale limitations that prevented its use in the larger scale multiparty authentication sections.

OPNET Modeller version 14.5 was used in the later research models to develop the multiparty authentication model in chapter 5.

OPNET Modeller is one of the most powerful simulation tools to represent distributed system architecture supporting a variety of application services, communication devices and protocols (Sahlin et al. 2015). It accelerates the R&D (research and development) process for analysing and designing protocols, and applications. It has the capability for application testing for Cloud systems.

It is a very effective, wide and accurate network modelling and simulation tool, and is the only tool having model libraries of real world network components and links (Guo et al. 2007). Given these advantages, a network architect can create architectures and topologies similar to real world networking environments, and use the results for defining strategies for real world network deployment projects (Guo et al. 2007). Given the significant costs involved in implementing such strategies, OPNET Inc. has made the tool very accurate and capable of modelling all the seven layers of OSI such that no aspect of designing is left out. These key advantages have been considered for choosing OPNET for this research.

4.2.1 OPNET Architecture

OPNET is a discrete event simulation tool that simulates a number of events occurring on the network at the repetition of a clock cycle. The object library is a collection of configurable network devices and links (both wired and wireless), preconfigured network devices based on products of leading manufacturers, network connectivity and routing protocols, configurable applications, and voice and video configurations employing commonly used CODECS and formats. The profile configuration for applications helps in generating traffic. All the configurations are carried out on an object palette after importing objects from the object library. Each object needs to be configured separately using attributes configuration screen. Normally, all configurations are GUI based, but advanced commercial versions support programming interfaces, as well. The simulator is built over the objects taken from the object library, object attributes configured in the model, and statistics chosen in reports. The simulations run in a separate discrete event simulation module showing the number of events in progress and the simulation speed in number of events per second. It also reflects the simulation errors, if any. The OPNET academic version simulates up to 50 million events, which is sufficient for this research. The reports are displayed in a separate reports module in as-is format as well as transformed formats into various statistical distributions. The plots can be viewed in continuous curve form, scatter plot form, bar chart form, and histogram form (Guo et al. 2007; Aboelela, 2003).

4.2.2 Simulation Projects in OPNET

The following process flow for managing a simulation project in OPNET.

Figure 14: Simulation project management process flow

The network modelling is done based on inputs about user and network data sets captured as a part of requirement specifications. In a research project, the requirement specifications evolve from the objectives and are configured with the help of known statistics from literatures and professional reports. These inputs are helpful in defining the attributes at each network object (nodes and links), protocols, traffic profiles, application configurations, and application profiles. If an object of desired type and nature is not present in the object library, the designer may have to create it first and then apply the attributes. These attributes need to be configured very accurately for ensuring that the network model works and traffic is generated. For example, configuring IP addresses with different network addresses on either sides of a link will not establish a PPP (point-to-point protocol) connection. Similar problem may arise if the bit rates at either side of a link are different or speeds less than 10 Mbps are configured on Ethernet links. After completing the applications and application profiles, they need to be applied on each node of the network for ensuring that the node understands what traffic it needs to generate or support.

After completing the model, the next step is to choose statistics from OPNET reporting engine. Statistics may be chosen as per the research objectives and the research questions/hypotheses. OPNET allows creating custom reports by grouping the statistics together. The discrete event simulator follows the attributes configured for each object and the statistics chosen by the modeller. Hence, OPNET does a focussed job as per the researcher's objectives and goals (Aboelela, 2003).

There are four models created in this study for fulfilling different goals. The scenarios created in the first three models are discussed in the next section. The fourth model is the outcome of this research.

4.3 **Description of Scenarios Created in OPNET**

In this section, the cloud models on cloud security, BI on the cloud, and BI security on the cloud are presented. The results of simulation are presented in the next section.

4.3.1 Cloud Security

The key security challenges and solutions on the cloud have been investigated in this section with the help of literature reviews and an experimental model created on OPNET that is simulated to produce useful statistics to establish the approach that the cloud computing service providers should take to provide optimal security and compliance. In this sub-section, a brief description of the multi-cloud model created on OPNET academic edition is presented. The following Figure shows the main interface to the model. The clouds shown in this interface are created using the IP network cloud objects in OPNET. An IP network cloud object can be expanded to enter another palette for carrying out detailed model comprising nodes and links. In the main screen, there are the following components:

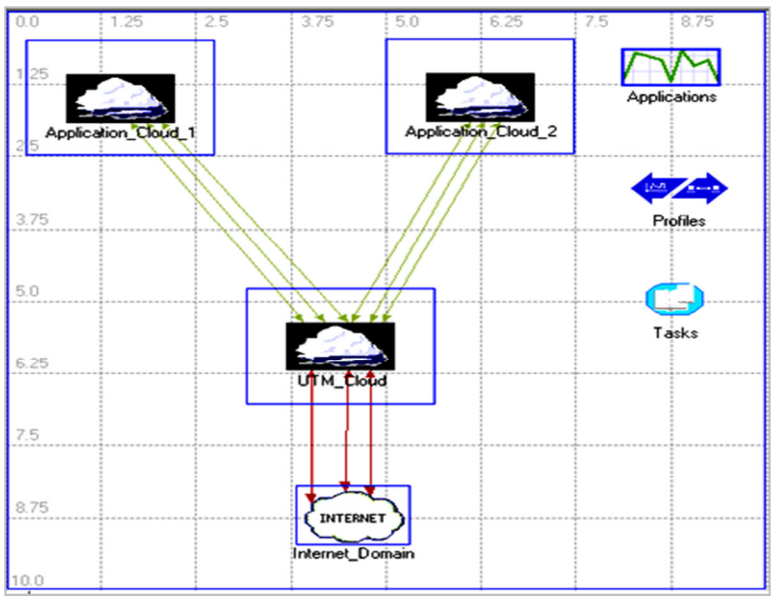

Figure 15: The main screen of the first model

a) The Internet domain: This is an IP network in which the backbone switches of an ISP (Internet Service Provider) have been modelled. The switches have been used to connect 1500 concurrent users of cloud hosted applications.

b) The UTM cloud: This is another IP network cloud object used to model a cloud infrastructure with security components only. In this model, the UTM (Unified Threat Management) cloud service provider is the primary interface between the cloud users (corporations) and the application clouds. The UTM cloud comprises all the security components required to protect the user networks and the application clouds from Internet based threats. The UTM cloud is explained in more detail later in this section, immediately after introducing its screenshot.

c) The Application clouds (1 and 2): The application clouds are IP network cloud objects comprising application server arrays and database server arrays connected to a cloud network. The Application clouds are explained in more detail later in this section, immediately after introducing their screenshots.

d) All the clouds are inter-networked using high end enterprise class switches with ATM OC-48 links. The servers are connected to the switches through 1000BaseX links. Hence, all switches possess ATM LAN Emulation enabled.

e) Application object: There are seven applications configured with built in load parameters as per default values in OPNET:

Cloud application – a HTTP browser based application with light browsing load

RDBMS Services – a high load database service

Antivirus and Antispyware application – a medium load database service

Anti-spam application – a medium load database service

Web services firewall application – a high load database service

LDAP services – a low load database service

Overheads – Encryption overheads configured as a custom application

f) Task object: The custom application for the encryption overheads has been created with the help of the task object. The encryption overheads have been configured as mild background traffic (1 KB to 4 KB per second) with fifteen to twenty packets delivered in one second between the source and destination using DDP (direct data placement – RFC 4296) protocol. The sources are the three corporate LANs and the destination is the UTM firewall object. This is an arbitrary choice made to generate a finite encryption overhead traffic on the network. The author has configured six phases in the task object to create application instances. The phases have been triggered at regular intervals of every five seconds after the start of the first phase. The protocol selected is DDP, and hence the overhead will not generate any TCP traffic or sessions, but will throw packets to and fro between the sources and the destinations defined in the phases of the task. In real network as well, the encryption overheads are additional streams of packets that do not contribute to any useful TCP or UDP session. This is the reason for the choice of DDP in the tasks object.

g) Profile object: The profile object has been included to configure the behaviour of applications configured on the network. The applications have been configured to run concurrently with uniform distribution of packets. An exponential, normal or log-normal distribution has been avoided to keep the model behaviour simple. The start times of the applications are within 5 to 10 seconds from the start time of the profile. The start time of the profile is within 50 to 55 seconds from the start time of simulation. A sufficient time gap is needed to allow the network build completely by building of the routing tables at all network devices.

Figure 16 presents the application cloud object. The servers configured in this object are the LAN objects with multiple server nodes attached to a central backplane chassis. This type of server configuration is similar to a blade server having a centralised chassis and multiple processor cards serving as individual servers. Normally, the processor cards share the storage attached to the entire chassis (not modelled in this project because OPNET academic edition doesn't support advanced server modelling).

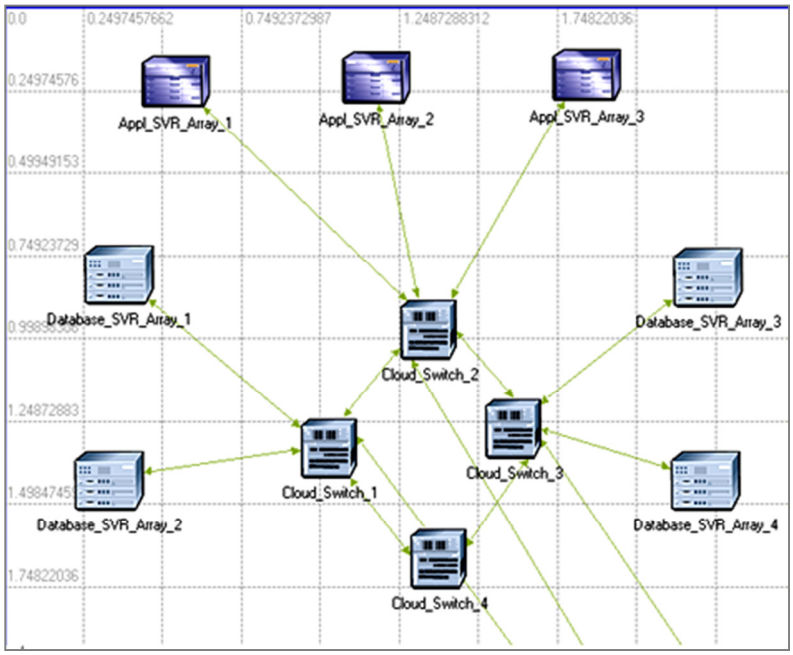

Figure 16: The application cloud object

The modelling of the LAN Object as a server array is presented in the following table. The number of processors is 32, shared among 25 workstations added in the LAN object. Given the huge backplane processing power configured in this object (attributes under red coloured rectangle), the LAN object serves as a LAN of servers and not workstations. Clouds comprise server arrays, and the chassis based servers (commercially known as blade servers) are best components to implement such arrays. The blades also comply with green computing requirement of clouds as they occupy very small rack space, consume significantly less

power, and dissipate much less heat, when compared with a similar array of standalone servers.

Table 1: The Modelling of the LAN object as the server object

Attribute	Value
Name	Appl_SVR_Array_1
Model	1000BaseX_LAN
Application: ACE Tier Configuration	Unspecified
- Application: Destination Preferences	None
Application: Source Preferences	
Application: Source Profiles	
- Application: Source Services	
CPU Resource Parameters	
- Number of Resources	32 No. of Processors
- Task Contention Mode	Simulate Contention
- Processing Speed Multiplier	5,0
- Multi-tasking Performance Table	
IP Host Parameters	
IP Processing Information	
- Processing Scheme	Slot Based Processing
- Backplane Transfer Rate (bits/second)	4,000,000,000,000
- Datagram Switching Rate (packets/second)	15,000,000
- Datagram Forwarding Rate	infinity
- Forwarding Rate Units	Packets/second
- Memory Size (bytes)	800,000,000,000
- LAN Background Utilization	None
- LAN Server Name	Auto Assigned
- Number of Workstations	25 No. of Servers in the Array

Figure 17 illustrates the Unified Threat Management (UTM) cloud. This is the interfacing cloud between the application clouds and the cloud users, with the UTM zone based firewall being the primary interface. The UTM firewall is an advanced stateful inspection firewall with multiple interfaces. In this model, three interfaces have been configured – users' interface, application clouds' interface and the De-Militarised Zone (DMZ) connected to multiple security servers, viz., UTM LDAP server for authentication and authorisation, UTM spam filter, UTM antivirus cum antispyware server, and UTM web services firewall comprising built in intrusion prevention capabilities (like Checkpoint).

Ideally, the author wanted to create server arrays for all these services, but the model was getting too heavy to generate useful results within the maximum cap of 50 million events.

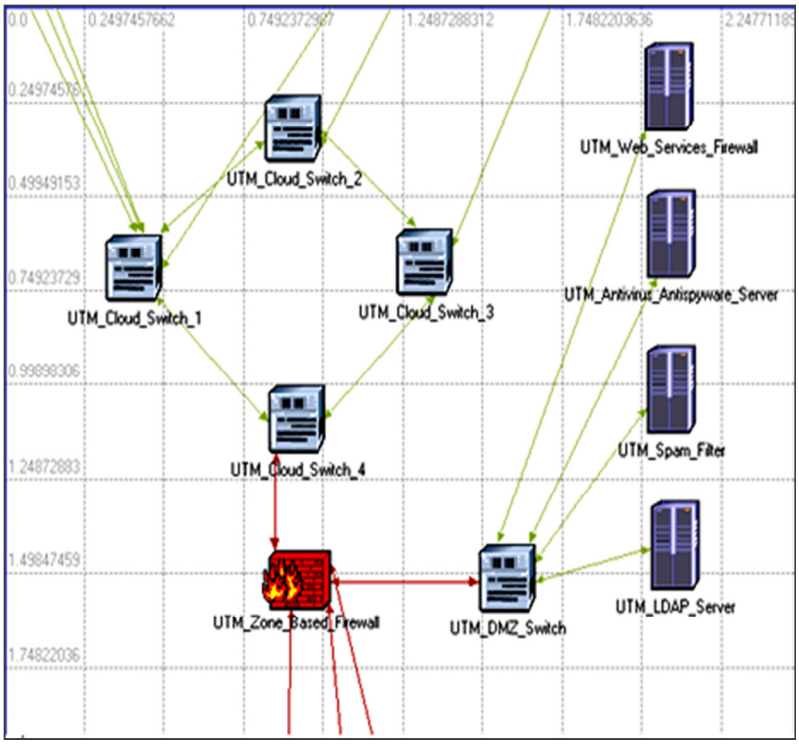

Figure 17: UTM cloud components

Hence, their poor response times in the results will have to be ignored. Each server has been tied with its own application by configuring its “application supported profiles” and “application supported services”. The UTM firewall is configured to support “overheads” to simulate the encryption overhead traffic. It forwards all the traffic to the security servers and the security servers in turn forwards the traffic to the application cloud servers. This has been configured by defining the “application source and destination preferences” in the attributes of the servers. As reviewed in the literatures, each security services server works like a database server. For example, the UTM web services firewall comprises a database of exploit signatures and blacklisted/malicious URLs, whereas the anti-virus/anti-spyware server comprises a database of virus and spyware signatures.

The cloud users are configured as LAN objects comprising 500 workstations each. Overall, 1500 workstations have been configured in the LAN objects. To generate the encryption overhead as the task phases based on direct delivery protocol, the user LANs and the firewalls have been configured as sources and destinations, respectively.

4.3.2 BI on the Cloud

The Cloud hosting of BI has been demonstrated with the help of a simulation on OPNET comprising a Cloud model with multiple OLAP application servers applying parallel query loads on an array of servers hosting relational databases. However, how do current BI software and online analytical processing (OLAP) frameworks perform in business environments, and how can BI be implemented on Cloud and how can its performance be measured. Figure 18 illustrates the main screen of the second OPNET model. The model comprises two large domains – the BI on the cloud domain and the Extranet domain comprising six corporates having 500 OLAP users in each corporate.

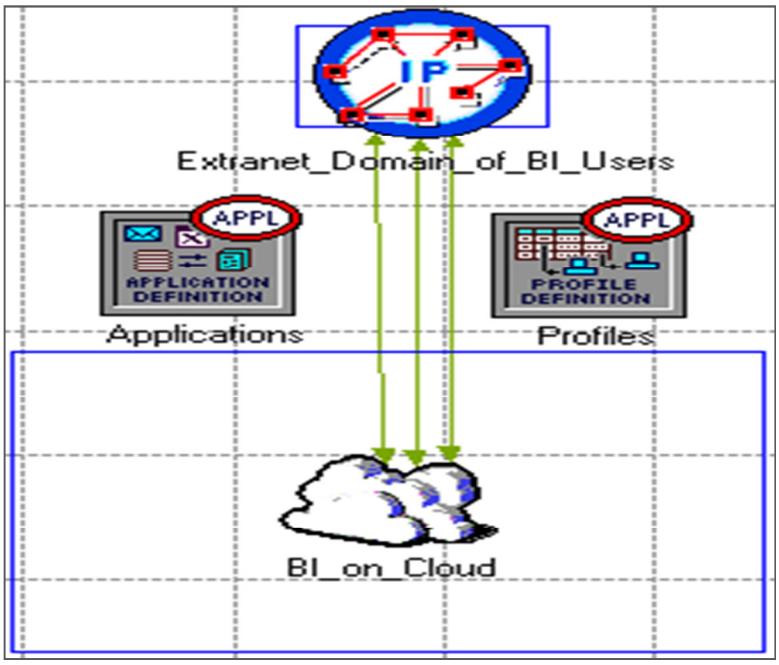

Figure 18: The main screen of the second model

The BI on the cloud domain is expanded in the figure below. The cloud has been formed using four nos. of Cisco 7609 high end routing switches connected in such a way that they can distribute the load evenly. The cloud switch 4 is dedicated to route all inbound traffic to the servers and send their responses back to the clients. The cloud switches 1 and 3 are serving four RDBMS servers each and the cloud switch 2 is serving 4 nos. of OLAP application servers.

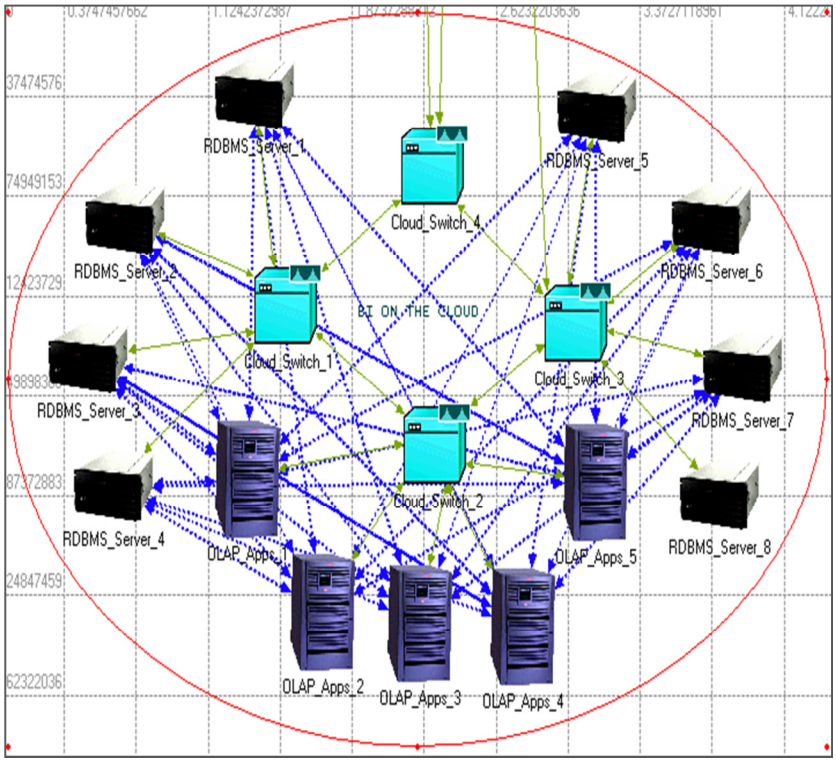

Figure 19: The BI on the cloud architecture

The blue dotted lines indicate the traffic flow distribution from the OLAP application servers to the RDBMS servers. As shown in the figure, the load from the OLAP application servers are evenly distributed among all RDBMS servers. The client load is routed to the OLAP application servers using destination preference settings on the client objects configured in the extranet domain, as shown in the below Figure.

The extranet comprises of three ISP gateway switches serving six corporate LAN segments having 500 users each. Each LAN objects have the four OLAP servers configured as destination preferences for the OLAP application profile. In this way, the OLAP requests from the clients are routed to the four OLAP servers and the RDBMS requests are routed from the four OLAP servers to the eight RDBMS servers (serving as a small scale server array in this model).

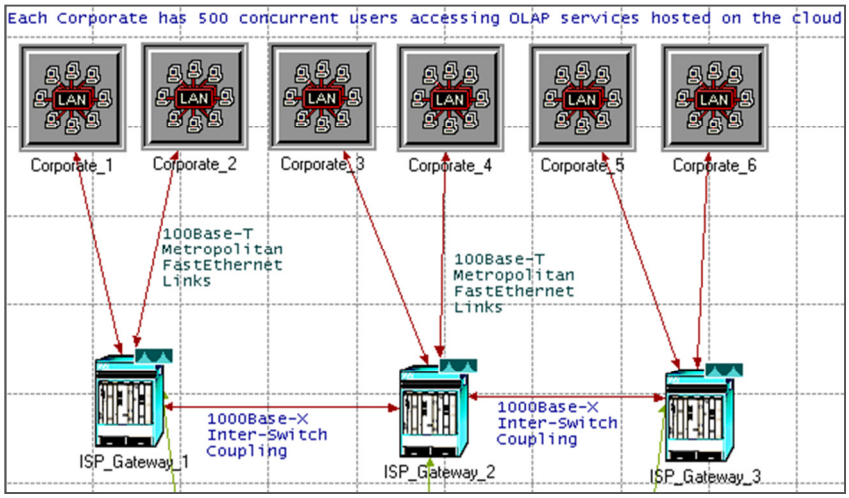

Figure 20: The Extranet domain comprising six corporates having 500 OLAP users in each corporate

The RDBMS queries are configured using the attributes shown in the following table. The default configurations of heavy database load of OPNET has been chosen and then increased by 10 times in shown in the Table below. This is based on the learning from the literature review that OLAP query load on databases is at least 10 times heavier than the normal query load. Moreover, the inter-arrival time of query has been set at 1 seconds and the type of service has been set at “excellent service”. Finally, the transaction mix of queries versus total transactions has been set at 100%. This is because the BI and OLAP framework does not have any data entry load because the framework is used for strategic decision support.

Table 2: The Database query settings to emulate OLAP query load on the databases

Attribute	Value
Transaction Mix (Queries/Total Transactions)	100%
Transaction Inter arrival Time	None
Transaction Size (bytes)	None
Symbolic Server Name	Database Server
Type of Service	Best Effort (0)
RSVP Parameters	None
Back-End Custom Application	Not Used

Table 3 illustrates the OLAP application has been configured as a heavy browsing HTTP application having varying 5120 bytes to 10240 bytes of object downloads per second (continuously updating dashboards), 7 to 10 objects per screen (dashboards, its description screens, legends, text boxes etc.), 1 second object refresh time (because the transaction inter-arrival time on the databases is 1 second) and 10 second page refresh time (ensuring that the OLAP screen refreshes after every 10 cube refreshes such that the user gets noticeable data changes at every screen refresh).

Table 3: The OLAP application profiling

Attribute	Value
Name	Profile
Profile Configuration	
- Number of Rows	1
Protocol_Tasks	
- Profile Name	Protocol_Tasks
Applications	
- Number of Rows	2
Protocol_Tasks	
- Name	Protocol_Tasks
- Start Time Offset (seconds)	Uniform (5,10)
- Duration(seconds)	End of Profile
Repeatability	
SAC-DB	
- Name	SAC-DB
- Start Time (seconds)	Uniform (5,10)
- Duration (seconds)	End of Profile
Repeatability	
- Operation Mode	Simultaneous
- Start Time (seconds)	Uniform (100,110)
- Duration (seconds)	End of Simulation
Repeatability	Once at Start Time

The application profiling of OLAP application (OLAP requests) and the RDBMS services is shown in Figure 18. Both of the profiles trigger concurrently with an offset of 5 to 10 seconds after the start time. The start time has been configured at 50 to 55 seconds to

ensure that all routing updates are successfully completed on the network before the application services are triggered.

4.3.3 BI Security on the Cloud

It is recommended that BI security model on a Cloud should comprises of network, transport, session and presentation layers of security controls through UTM, and application layer security through the distributed security components. In this section, two models for securing BI on a cloud have been simulated. The first model is based on securing BI using a Unified Threat Management (UTM) cloud and the second model is based on distributed security controls embedded within the BI server arrays deployed throughout the Cloud.

This scenario is divided into two models – Model A and Model B. Figure 21 presents the Model A of BI security on the cloud. The model comprises three large networks – an extranet domain of BI users, the UTM cloud, and the cloud hosting BI services.

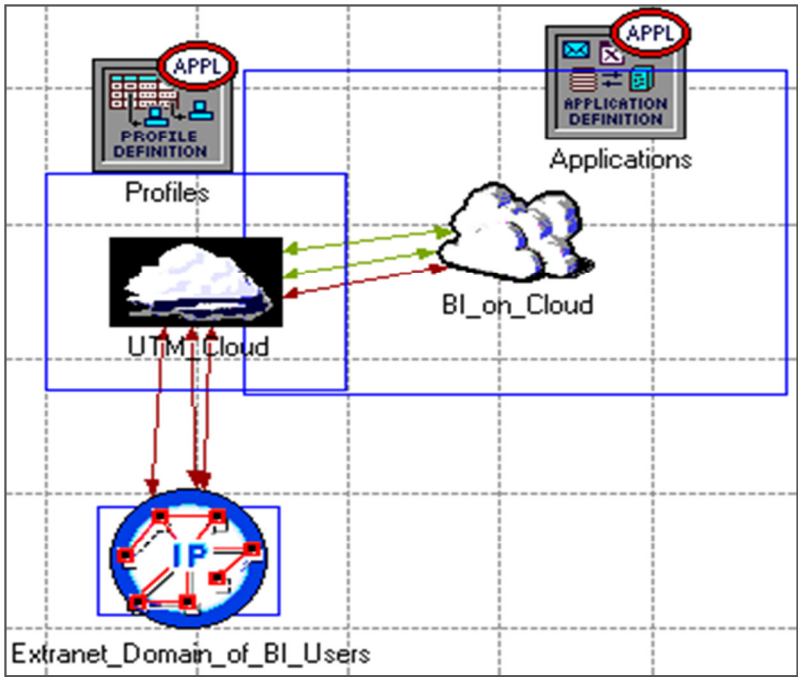

Figure 21: The Model A of BI security on the cloud comprising access of BI users through a UTM cloud offering security-as-a-service

The BI cloud comprises two server arrays as shown in the below Figure. The DW_DM servers are the data warehouse/data mart servers hosting the temporal and non-temporal databases. The OLAP apps servers are OLAP application server arrays hosting the OLAP dashboards application, and the temporary views to create and transform the OLAP cubes. The servers are connected through an array of cloud switches formed by Cisco 7000 series switches. The hardware chosen in the model are also changed and all application demands have been eliminated. The DW_DM servers chosen are high-end Dell servers and the OLAP servers chosen are standard HP servers from the OPNET objects database.

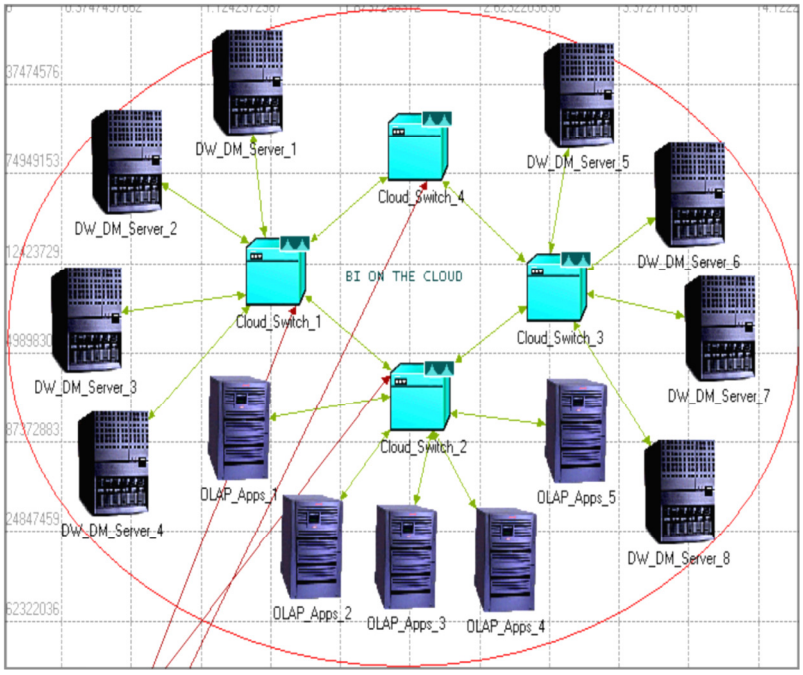

Figure 22: The BI server arrays forming a cloud infrastructure

Table 8 (Appendix B) illustrates the OLAP application and data mart/data warehouse services modelled in OPNET, along with the security services. The OLAP and database modelling has been done as per two earlier models (Cloud security and BI on the cloud). However, the security services modelled in the table 7 are based on reviews conducted in this study. The services modelled are the following:

a) OLAP_DASHBOARDS: A heavy-load HTML application with large size objects (200KB; 15 nos. per page) and an interval of 3 seconds between two transactions

b) DW_DM: A heavy database application

c) UTM_DB-ACT_MON: A heavy load database application representing an activity logger and monitor for all database commits in the data marts and warehouses; this is similar to the system that creates embedded security related database objects (separate tables) comprising information as per the privacy metrics

d) OLAP_VIEWS: A medium load temporal database application representing OLAP views

e) ANTIMALWARE: A medium load database application representing anti-virus and anti-spyware software (checking all files against a database of malware signatures)

f) WEB_SECURITY: A low load database application verifying all web service calls and requests against a database of users and privileges

g) ANTISPAM: A medium load database application verifying all SMTP transactions against a database of spam signatures

h) IDPS: A high load database application tasked to verify all incoming packets against a database of exploit signatures such that intruder sessions can be detected and blocked

i) LDAP: A low load database application that takes authentication and authorization requests and compares against a database of user identity details and privileges; it also maintains an account of all successful and failed requests

All of these applications have been grouped under three types of application profile, each having a different role in the system. The profiles are required in an OPNET model to

apply on server systems, selectively, such that the role, traffic load, traffic patterns, initialization and termination, and inter-session delays can be clearly defined. The profiles also help in defining a traffic pattern initiated by a server or a client-end device, like – constant, linear, logarithmic, serial-random, serial-ordered, parallel (with overlapping times), concurrent, or exponential. If Quality of Service is implemented, the profiles also help in defining traffic prioritization. For most of the commonly used applications, OPNET provides configuration windows. However, custom traffic profiles can also be created based on observations made in a real world application environment.

The profiles configured in the two models are shown in Table 9 (Appendix B) as following:

Three profiles have been created in the models – BI_Security_UTM, BI_Application, and BI_DW_DM. The BI_Security_UTM has the security and privacy related services grouped under it, the BI_DW_DM comprises the DW_DM application, and the BI_Application comprises the OLAP_DASHBAORDS and OLAP_VIEWS positioned under it. The purpose of creating three separate profiles is to assign them independently to the hardware marked for the three roles in the first model. Hence, all hardware under the UTM cloud was assigned to the profile “BI_Security_UTM” with application services enabled as per the role. The connectivity of all users to the BI cloud is routed through the UTM cloud. The destination preferences of the user machines are the UTM servers, and those of the UTM servers are the cloud servers. In this way, a user machine cannot contact the cloud server without its traffic being routed through all security servers in the UTM cloud. The user cloud comprises six corporate LANs with 500 workstations in each LAN. The workstations are connected to ISP core formed by Cisco 7000 switches those are uplinked to the UTM cloud using 10G Base-T links. The BI and the UTM clouds comprise switch arrays interlinked with

ATM OC-12 links. Hence, bandwidth is not a constraint given that the clouds need to serve only 3000 concurrent users.

The Model B of BI security on the cloud is presented in the figure below. In this model, the UTM cloud is eliminated and all user workstations are directly interfaced with the BI cloud switches. Given that the UTM servers have been deleted, the profiles and their application services need to be redistributed among the remaining servers – the OLAP apps servers and the data marts and data warehouse servers. The following distribution has been implemented:

a) BI_DW_DM: the data warehouse and data marts servers

b) BI_Application: OLAP apps servers

c) BI_Security_UTM: all the security services are enabled on all the servers (with an assumption that each server will need the services of these security components).

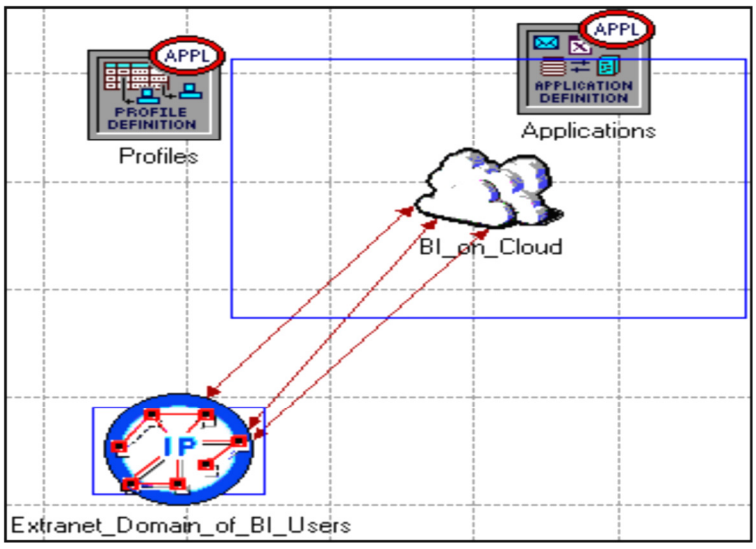

Figure 23: The Model B of BI security on cloud computing with the UTM cloud eliminated and all users directly connected to the BI application servers on the cloud

Hence, this setup represents the distributed embedded security concept for BI advocated by a number of scholars, as reviewed in Chapter 4. The application destination

preferences have been changed to point towards the OLAP apps servers, which in turn has their destination preferences set as the data warehouse and data mart servers. After running the simulation, the following result is obtained. It may be observed that the simulation time has reduced to one minute from two minutes because the number of events has increased on the network (OPNET academic edition supports a simulation up to 50 million events only). The increase in number of events per second is because the security-related services are no longer confined to their respective servers albeit are distributed across all the servers in the two arrays.

4.4 **Simulation results and Discussion**

In this section, the simulation results of the three cloud models on cloud security, BI on the cloud, and BI security on the cloud are presented. The discussions are referred to the screenshots presented below.

4.4.1 Cloud Security

Figure 24 illustrates the application response times. The database traffic comprises of the security server services as well as the database services for the cloud based applications.

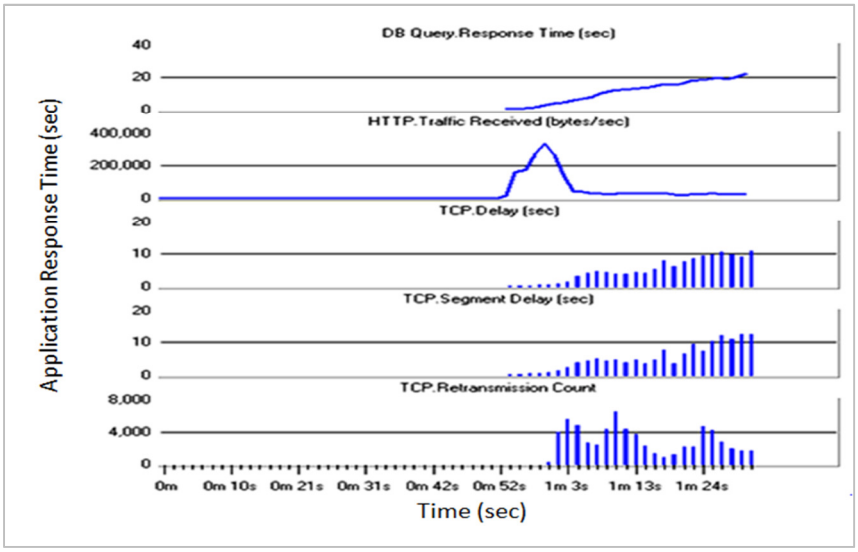

Figure 24: Application response times

Figure 24 indicate that there are negligible queuing delays on the ATM links (given that they are OC-48 connections), but the application performance is very poor. No user will accept 20 seconds of response time of a DB query and 10 seconds response time of an HTML object download, although the application traffic has been configured at light browsing load and the database traffic has been configured mostly at medium load. This is because the UTM cloud is the bottleneck. The response times are low in this model because all the traffic is first forwarded to the security services servers (for necessary inspection and clearances) and then delivered to the application servers. The author has configured standalone servers for security services in the UTM cloud and hence there are congestions.

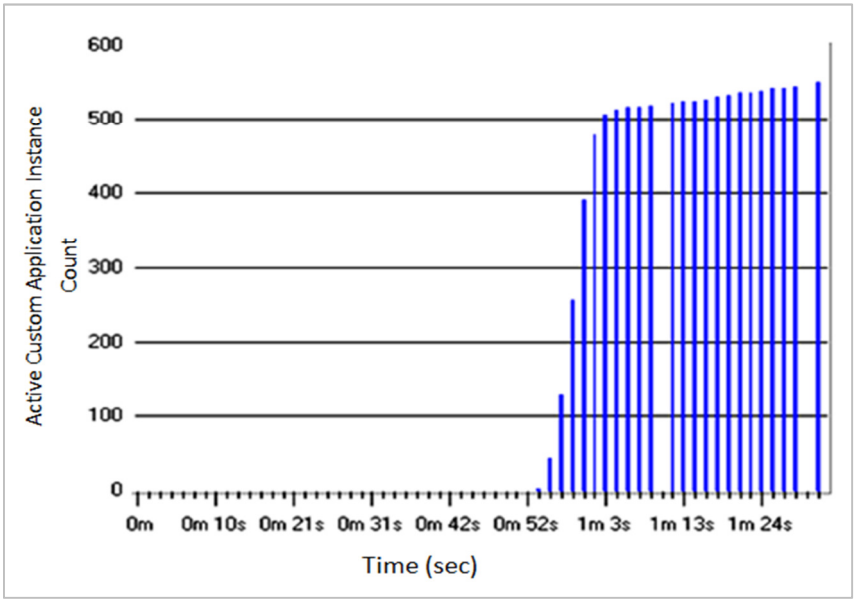

Figure 25: A sample of overhead requests count from one of the user LANs indicating the encryption overhead using direct data placement (DDP) protocol.

The http traffic is the browser based traffic of the cloud applications. In addition to these traffic statistics, the simulation has also captured the overhead traffic shown in Figure 25, comprising encryption overhead modelled as direct delivery protocol requests/responses.

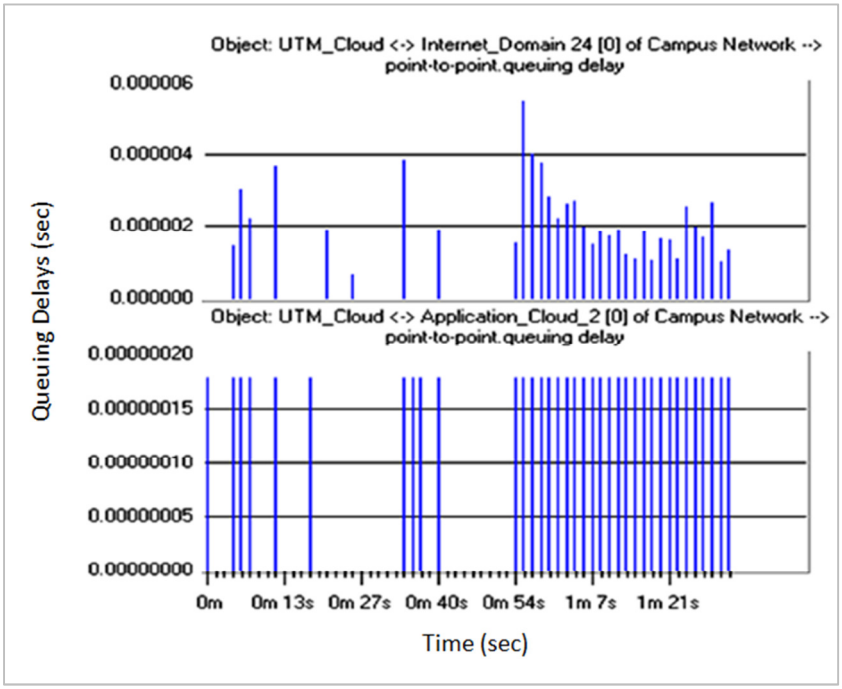

Figure 26: A sample of queuing delays between two inter-cloud links

However, will the situation be different in real world clouds when the security servers are implemented as large scale arrays? How many firewalls (serving as VPN concentrators) will be required? Will the UTM providers implement large scale firewall arrays as well? A user company will hire the services of only one UTM provider to connect to the application clouds. This means that there will be fewer UTM cloud service providers than application cloud service providers. The cost of implementing large arrays of security servers and firewalls will be very high. Hence, at some stage, the resources on demand (elasticity) of the application clouds will suffer due to bottlenecks at the UTM clouds. The users may have to maintain two SLAs – one with the UTM provider and other with the application services provider. Hence, when the response times degrade, the user organisations will have to negotiate with two different parties, and it will be very difficult to ascertain where the problem is. But in such a scenario, it may be very difficult for the user company to maintain appropriate security and governance of the resources maintained by them on the cloud. The

UTM provider may take accountability of external security threats, but once the traffic has reached the application cloud, they will be out of this obligation. In such an arrangement, if there is a security incident at the application cloud, the user company will find it difficult to identify who is accountable – the UTM provider or the cloud application service provider. Hence, based on the results of the simulation and the problems evident thereof, the dual party model comprising unified threat management clouds separated from application clouds may not be effective in cloud computing threat management, and resulting risk mitigation although inter-cloud performance is not an issue. The application cloud providers will have to launch their own UTM services, or else change the architecture to distributed threat management, like – security services embedded within the application and database servers deployed in the arrays, and the cloud switches acting as firewalls and intrusion prevention devices. For example, it is possible to configure a Cisco router as a firewall and intrusion prevention device. The mechanism of distributed network admission controls, and authentication/authorisation will have to be implemented. The author would like to present another example. There should be some mechanism to build the LDAP services within the core database server array in some kind of multi-tenancy configuration. In this configuration, the database objects (tables) defining the multi-tenancy attributes should also comprise the parameters configured in an LDAP server. In this way, the LDAP services will be built within the database servers meant for cloud applications, and separate LDAP arrays will not be required. Similar mechanisms need to be invented for anti-spam, anti-virus/antispyware, web-services firewalls, etc.

The security architects should come out of the “in-the-box” mindset and spread security solutions across the components of an application cloud. The author will prefer to call it “embedded UTM” within application and database servers of the application cloud. In this model, there will be single point of accountability from the users’ perspective, because

the cloud application provider will also be accountable for data protection and security of the client resources, and the user organisations will find it easier to carry out risk management and mitigation practices. This will also result in optimum performance of the cloud applications, and added revenues for the cloud based application service providers.

4.4.2 BI on the Cloud

The results of simulation are presented in the below figure. The query load is not exactly the same on the RDBMS servers but the pattern indicates almost even distribution of query load. This is evident from the "database query requests per second" statistics collected from the eight RDBMS servers are stacked one above another as shown in Figure 27.

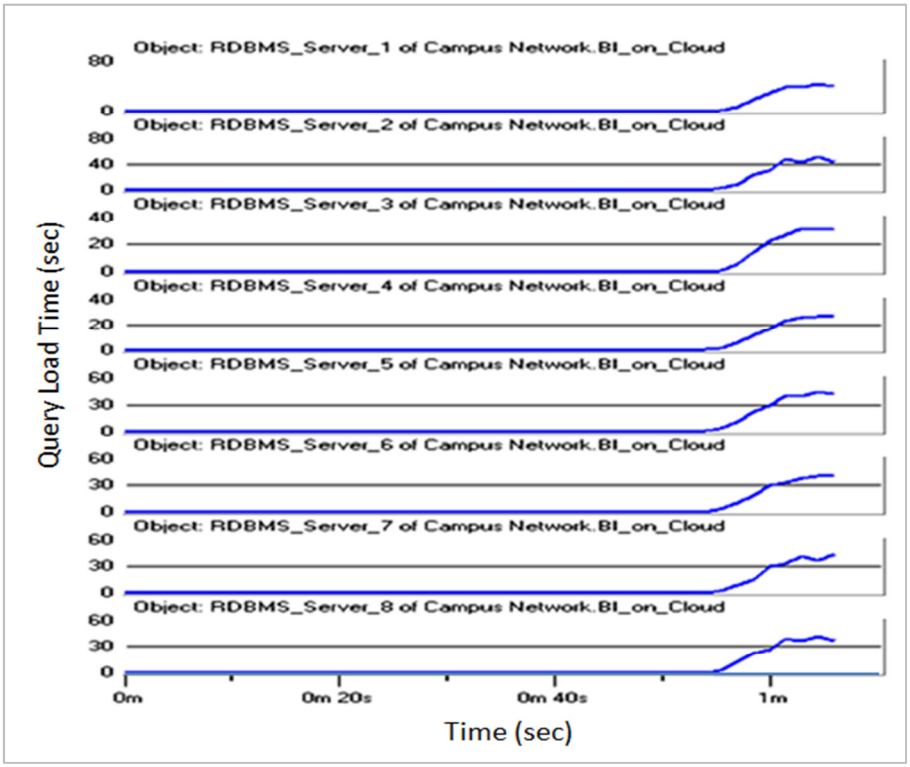

Figure 27: Query load on the RDBMS servers

The query load is slightly above or below 40 requests per second on all the RDBMS servers. This reveals that the load distribution through appropriate network configuration and application demand.

These configurations have caused near even distribution of query load from the four OLAP servers on the RDBMS servers. Moreover, the query task processing time on the database servers are also nearly even as shown in Figure 28. This has been possible because the author has deliberately chosen the same hardware make, model and configurations for all the eight RDBMS servers.

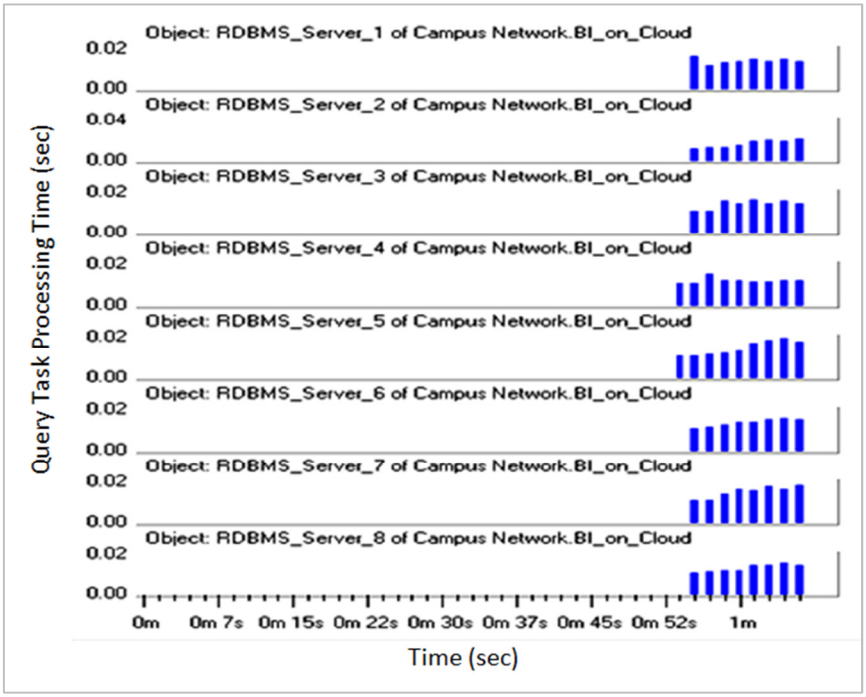

Figure 28: Query task processing time by the RDBMS servers

These results are a good demonstration of how a massively parallel RDBMS system can be deployed to form a BI and OLAP framework and how the framework should perform in the cloud environment. This is in line with the requirements stated by scholars as reviewed in the literature study. However, there are a few key points that should be kept in mind about this model as listed below:

a) First of all, the model has only eight servers in the RDBMS array serving only four nos. of OLAP application servers.

b) Secondly, the load distribution has been managed evenly through application demand flow modelling which is an excellent feature of OPNET and works very well.

c) Thirdly, the servers chosen in this model are of the same make and model having identical hardware configuration.

d) Fourth, the load has been modelled as constant after an exponential increase at the start. The simulation of the load carried out in this model has lasted only for fifty million events and with no load variations.

e) Finally, this model comprises only 3000 OLAP users connecting concurrently. A real BI environment on cloud computing will have tens of thousands of end users applying concurrent BI load on the servers.

These are ideal scenarios that would not be possible on the cloud. But these settings in OPNET have evolved the challenges that will be faced in moving BI to the cloud as per the requirements stated by the scholars. A cloud will have hundreds of servers in the arrays and hence even distribution of network load will be a very challenging task. The architects will have to watch for the bottlenecks on the inter-switch connections, even if they are deployed using the fastest possible ATM connections or the 10G gigabit Ethernet. The load distribution will have to be managed by advanced provisioning engines and routers, which will not be as easy as configuring application demand flow patterns in OPNET as indicated by blue dotted lines. These provisioning engines and routers need to be optimized to ensure that the user load is evenly distributed among the servers in the array and spilled over to additional arrays if there is an overloading scenario. Also, it may not be possible for the IaaS provider to implement a cloud with identical hardware make, model and configurations. Hence, the query

processing response time of each server will be different due to differences in hardware configurations. Hence, a mere even distribution of load to the servers by the service provisioning engine and the router will not serve the purpose. There should be some intelligence to route the load based on the knowledge of query processing response times of the servers. The servers with slower response times should get lesser load compared with the servers with faster response times to eliminate wait states at the receiving end. The capabilities of RDBMS partitioning, RDBMS load balancing, web provisioning application services, services routing engines and query performance optimizing should be exploited effectively by the BI architects to ensure that the massively parallel processing system of database server arrays works perfectly to effectively utilize the processing power of the servers and synchronize the query processing times to reduce/eliminate wait states at the application servers' end.

The above discussion presents one more challenge in taking BI to the clouds. The SaaS, PaaS and IaaS providers may be different companies. Hence, to ensure the above requirements of BI hosting on clouds, these providers need to carry out excellent coordination of architectural detailing for designing and deploying the services to enable the various layers of BI and OLAP framework. BI cannot be implemented in an ad-hoc way by the providers otherwise it will suffer from the same level of bottlenecks and resource crunch as it has been suffering in the self-hosted environments. The providers need to carry out effective planning of every detail and implement the infrastructure components, platform components and application components to achieve a true massively parallel processing system with highly elastic capacity enhancement framework using all available technologies efficiently.

4.4.3 BI security on the Cloud

The performance of Model A is presented in the figure below. The average response to database queries on the entire network is between 20 to 40 seconds and the average http object response time varies between 1 to 3 seconds. The TCP delays have exceeded 20 seconds, TCP segment delays are between 2 to 4 seconds and the TCP retransmission count has exceeded 1000 twice during the simulation. The performance degradation is clearly due to capacity crunch. There are three factors affecting the performance of this network.

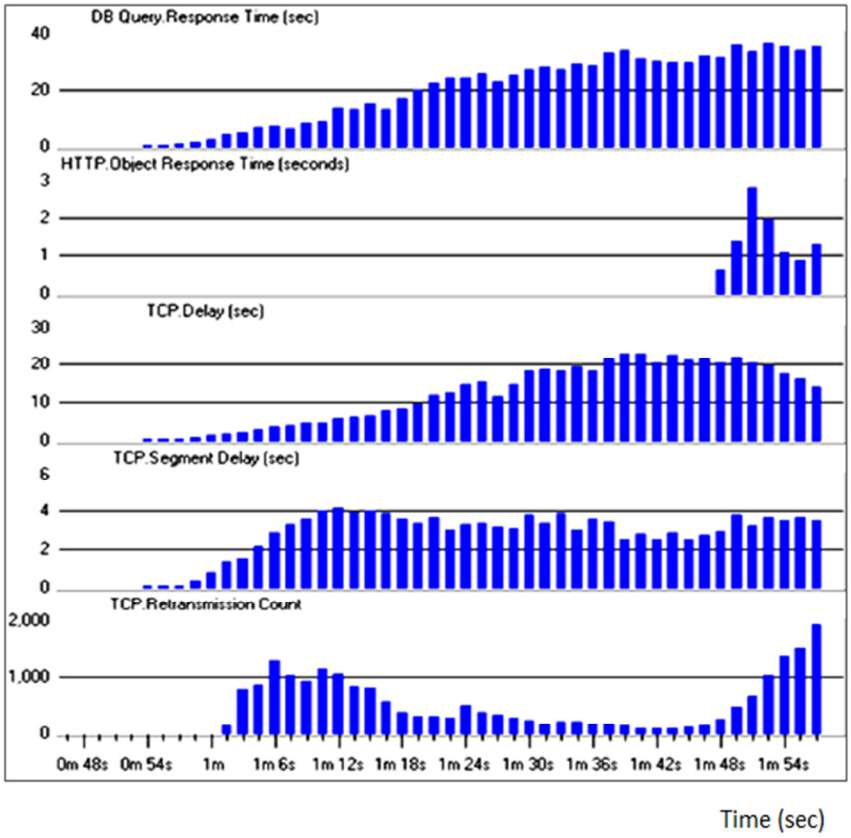

Figure 29: Performance of Model A of BI security on the cloud

All the user traffic is being routed through the security services servers in the UTM cloud. The users are connected to a zone-based firewall, which routes their traffic to the security servers. In this way, there are two hops from the user workstations to the BI application servers on the cloud. In this arrangement, the capacity may not be an issue but the

routing of traffic through the servers placed in the De-Militarized Zone (DMZ) may be adding a delay component on all the servers.

The security servers are adding significant overheads over the user traffic in the process of carrying all the checks and verifications configured in the model. One may view the security server arrays as one large server object inspecting two other major server objects (OLAP and BI) over a high-speed network interface. These servers are facing each other as single entities with large number of TCP sessions consolidated into the four network uplinks. Hence, TCP delays and retransmissions are occurring on the network.

With a large number of TCP session requests processed by the security servers on the UTM cloud, congestion at the network and transport layers is evident. The congestions are not due to data-link layer problems (because all links are either 10G Base-T or ATM OC-12 high capacity and high bandwidth optical fiber channels). The congestions are clearly due to limited TCP session handling capability of the security servers in the UTM cloud. There are two roles of each security server – (a) to inspect all session initiation requests emanating from the end users and forward the authenticated ones, and (b) to establish separate TCP sessions with the OLAP and data warehouse/data mart servers on the cloud for monitoring and logging their transactions.

The performance may degrade further if the number of user workstations is increased, resulting in a rise of TCP sessions through the uplinks to the UTM cloud, and increasing the number of transactions per server (thus increasing the monitoring and logging overheads). The embedded security-related database objects comprising details of transactions may add to a huge archives requiring separate management. Hence, there should be a way to purge older activity logs in the system, automatically.

The performance report of Model B is presented in the below Figure. In this model, the performance is within the expected limits and there are no TCP delays, TCP segment delays and TCP retransmissions. It is observed that the performance of database query response and html page and object responses is much better than the first model and is very satisfactory from a user's perspective.

The response times returned in this model are the ones expected from a cloud hosting of BI. Elimination of the UTM cloud layer has ensured that all the session bottlenecks in the network are gone.

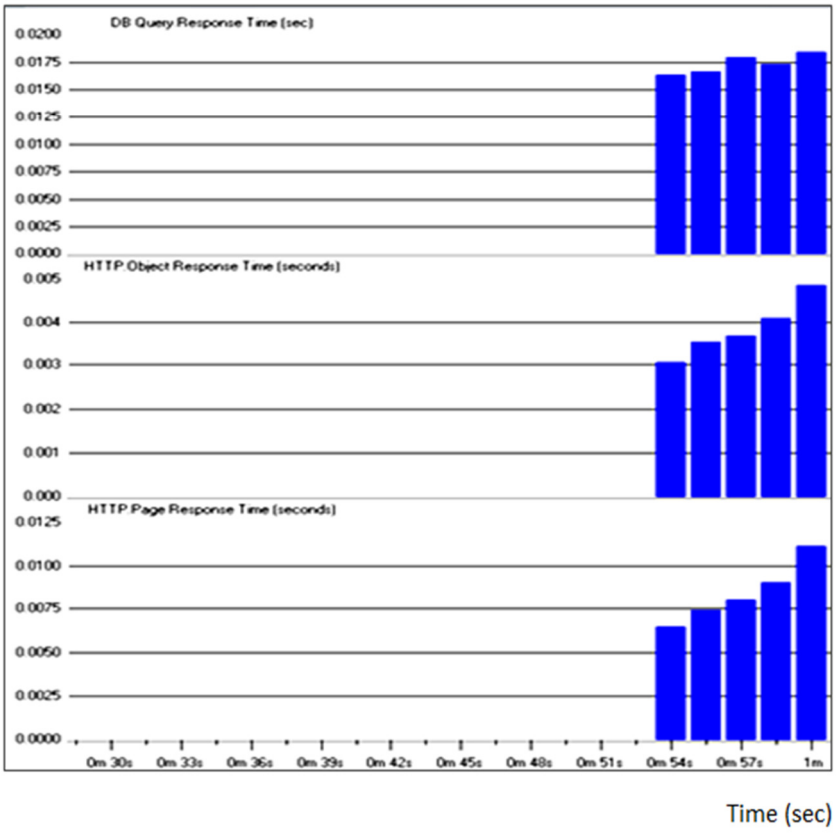

Figure 30: Performance of Model B of BI security on the cloud

The user sessions are served by the BI cloud based server arrays for application requests and for the security sessions, as well. The load of the security sessions are distributed among all servers in the OLAP and data warehouse arrays given that the databases

serving the security applications will also be partitioned and spread across the servers. Hence, the massively parallel processing model proposed by Al-Aqrabi et al. (2012) holds good for security services as well. This is essentially a practical form of implementing the distributed embedded security components recommended by the scholars as evident from literature review findings.

What will be result if the UTM cloud owner provides significantly large sized arrays with servers plugged to them, and partition all security related databases to produce a massively parallel system? The performance at the UTM cloud will definitely improve, but the network and session layer congestions cannot be reduced. This is because the system will be a cascade of two large clouds, and the cascade itself will be a bottleneck. The effects of a massively parallel database query-distribution system will not work when two large arrays are cascaded by joining two clouds. This limitation will be there when two clouds are interconnected and the user sessions will be allowed to pass through one cloud to the server arrays of another. The performance will definitely improve when the users are allowed to connect to two clouds independently for different purposes, and there is no inter-cloud cascade. However, such a system will not be effective from security point of view, because the user sessions need to pass through a common checkpoint before allowed accessing the application resources.

4.5 **Discussion**

In chapter 2, the author reviewed cloud computing security with special emphasis on governance of the security and compliance from the perspective of user companies as well as cloud service providers. A number of literatures have been reviewed to present what can be done to make the cloud hosted businesses secure, reliable, compliant and long lasting. The NIST recommendations and the supporting literatures have been taken into account. Some

literature highly recommended that the cloud security should be hosted as a service oriented framework and the accountability should with a separate security-as-a-service provider. The NIST recommendation on risk assessment and compliance also becomes quite effective in this model because the risk transfer and risk avoidance can be carried out effectively by handshaking with a specialist cloud service company rather than application cloud providers that may undertake security as an additional responsibility. However, a report by Gartner recommends that virtualisation security cannot be implemented in centralised manner following the UTM approach. The simulation results of first scenario presented in this chapter tend to support Gartner recommendations. However, the Gartner report doesn't talk about cloud security and hence this report may be one of the very few that views UTM from the other side of the table.

Cloud computing comprises three ways of provisioning services – software-as-a-service (SaaS), platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS). These services may be provided by same or different providers depending upon the business arrangements. However, the SaaS provider needs the settings on the PaaS and IaaS clouds to be defined as per the application services provisioned through the web services architecture components. Clouds comprise the service provisioning and routing engines that can effectively sense the loading pattern on the underlying resources.

BI and OLAP framework is highly resource intensive. It has a multilayer architecture comprising multidimensional OLAP cubes with multiplexed matrices representing relationships between various business variables. The cubes are formed by sending OLAP queries to the data warehouses stored in the RDBMS servers. The size of an OLAP query is typically 10 to 12 times larger than an ordinary database query. Hence, if BI and OLAP framework is taken to the cloud for serving hundreds and thousands of end users, it is essential that the cloud providers implement massively parallel processing RDBMS systems with even

distribution of query load and query response times for the OLAP application servers. In this study, a BI and OLAP framework has been modelled using OPNET and the requirements of a massively parallel RDBMS server array has been modelled using the OPNET features. The results have reflected the ideal scenario for taking BI to the cloud. However, the real clouds will not have ideal configurations as made in this OPNET model. Hence, the real challenges on the cloud needs to be identified and addressed to ensure that the results can be brought closer to ideal scenario as far as possible (Al-Aqrabi et al. 2012).

The details of challenges in implementing a massively parallel processing RDBMS server system to take BI to the cloud have been discussed. In addition, the deviations from the ideal configurations in the OPNET model presented in the research have been discussed. The author has tried to discuss the solutions to address the challenges and the deviations from the ideal scenario of the OPNET model. In future, researchers may like to study modern technologies pertaining to service provisioning, service routing, schema partitioning, load balancing, etc. to implement an enterprise level RDBMS system to achieve a massively parallel processing RDBMS server system for taking BI to the clouds. In this context, there is a significant opportunity to carry out multiple experimental studies to evolve the practical configuration solutions useful for the cloud service providers targeting to host BI and OLAP framework on the cloud.

Before the security controls in a BI environment is discussed, a brief analysis of its security requirements is presented below (based on the findings of the literature review):

a) The security controls in BI is different from traditional database applications. The data units are extracted from various transactional and decision-support databases, which belong to a number of information owners. Although, the data units are extracted for BI reporting purposes only, their original ratings, as an information asset (confidentiality, integrity, availability, and sensitivity) cannot be discarded. Hence, it

is important that the details of information ownership and security related attributes of the original information units is captured in the temporary data marts, preferably with details of legal and regulatory compliance obligations.

b) The data transformation and modelling agents should be tasked to take care of the security details tagged with each information unit, and include appropriate summaries in separate tables along with the rest of the metadata entries. In addition, the identity of the transformation agents should be tagged along with the metadata of the data units after building the hierarchies, dimensions and multi-dimensional relationships.

c) After loading the transformed data into the data warehouses, the system should have features to capture monitoring and activity logs for all data units tagged with high security attributes. Separate objects should be defined to capture such details. For example, the Oracle enterprise security monitor constructs separate database objects to capture activity and monitoring logs, and generates automatic alerts and alarms as per the specified rules.

d) The data related to security attributes tagged with the data units and the activity/monitoring logs should be encrypted within the database objects.

e) The security controls pertaining to network security, session security, and transport security should be implemented to protect the BI framework. Some examples of such security controls are:

    a. Antimalware: comprises antivirus and antispyware tools
    b. Anti-spam: for SMTP protection of inbound and outbound automatic mails in the BI framework
    c. Intrusion detection and prevention systems: for detecting and blocking exploit attempts

d. Web services security: to protect all components of web services in the BI framework
e. LDAP server (alternatively, RADIUS and TACACS servers): for providing authentication and authorization services for end users, database administrators, OLAP administrators, and the transformation agents

When BI is taken to the cloud, it takes the form of service-oriented architecture. The XML data files find a significant role as the data files of web based data warehouses or data marts, and as OLAP multidimensional cubes. Hence, the embedded distributed security needs to be implemented within the XML data structures, as a part of the document object models. The security related attributes, tagged with the data units can be embedded within the XML hierarchies, or can be part of separate relational tables marked as metadata tables, which can also be exported in the form of XML files. Any changes in the metadata information can be imported back into the database tables using DOM parsing. The transformation agents can use this feature to maintain metadata XML files, including the security attributes embedded in it.

The LDAP server on the cloud may also be a separate relational database server holding details of all authorized users, their authorization levels on the database objects and their access to OLAP views. Hence, LDAP should be viewed as a set of separate objects ensuring appropriate segregation between the user sessions. Privacy and trust are two significant issues on the cloud after a critical business application (like BI) is migrated to it. The user sessions should be segregated using trustworthy technologies, where LDAP, RADIUS and TACACS can play a significant role. The security policies may follow the cloud cube model. Cloud cube is an innovative representation of directing the security policies based on business needs of an organization (Chea et al. 2011). The cube presents four forms of security policies:

a) Internal and external model: In the internal model, a company may demand a physical boundary of data ownership, and in the external model, the company may be flexible to keep the data units outside the physical control of the data owners. The former can be implemented on a private cloud and the latter can be hosted on a public cloud. In a private cloud, a company may be the sole owner of the LDAP, RADIUS or TACACS engines, and all other security components mentioned in this document. In a public cloud, all the security components and the LDAP functionality will be shared among multiple companies implementing an appropriate, reliable and technically sound segregation mechanism.

b) Proprietary and open model: In the proprietary model, a company may demand hosting of proprietary software platforms and interfaces. In this mode, the distributed and embedded security components will also have proprietary features (examples are special-purpose hardware security appliances and firewalls, and Oracle and IBM relational databases and tools). In the open model, the software platforms and interfaces can be open (example, Google Apps) and the embedded security components may also be open (examples are Linux based firewalls and IDPS systems hosted on standard servers). In open model, a platform-as-a-service provider may be involved for offering platforms as a service (including the platforms for distributed embedded security).

c) Perimeterised and deperimeterised model: In the perimeterised model, the mission critical database and application services may be deployed within secure perimeter (like a De-Militarized Zone, DMZ) defined and controlled by a zone based firewall. The UTM cloud presented in this research is an example of a perimeterised security setup. In such a setup, the security components may be separately positioned in a DMZ like environment, or the critical application and database servers comprising

distributed security components may be separated from the rest of the components and placed within the DMZ. In deperimeterised setup, the controls may be placed just like the perimeterised system but no perimeters or boundaries are formed to identify and separate critical components from non-critical components.

d) Insourced or outsourced model: In the insourcing model, the security controls may be deployed within the cloud periphery owned by a company, and managed by internal employees. In an outsourced model, the security components may be outsourced to the provider offering security-as-a-service (example, a UTM cloud). In the outsourced model, distributed and embedded security components will not be possible to implement.

Based on the findings of the modelling and simulation experimentation in this project, and their mapping with the cloud cube model proposed by Chea et al. (2011), the following scenarios are possible:

a) Internal model: In this model, the OLAP servers and the relational database servers for data marts or data warehouses are hosted on arrays owned by a company as a part of their private cloud. The security components may be distributed, and embedded, and managed by in-house employees. The performance will be highly effective as revealed by the simulation results for distributed embedded security components. However, to sustain the performance, businesses will need to observe the increase in number of TCP sessions, continuously, and upgrade the server arrays. This model may comprise limited number of servers per array and may be very expensive given the high capital and operating costs of owning a private cloud and all the security components on it. Typically, banks or financial institutions may like to own a private cloud with internal model of security.

b) External model: This model may be applicable for any company that is flexible to use a shared cloud to host BI framework. Such a model may be highly useful to promote BI as a SaaS offering. The security components can be distributed and embedded and all security services may be shared among multiple tenants on the cloud. This setup will be very cost effective given that the cost of BI framework and the embedded security services may be available on pay-per-use basis. As per the simulation results, this model returns excellent performance results.

c) Proprietary and open model: This model will be based on the choice of BI and security platforms made by the companies. The proprietary platforms (especially the security appliances bundled in specially packaged hardware) may be more expensive and hence lesser number of devices will be deployed per array. Hence, cloud arrays with proprietary platforms may perform poorer than open platforms given that the company owning them may be under compulsion to make maximum use of the available capacity before investing in additional appliances per array. Open platforms comprises a number of open source appliances (like the Linux based zone based firewalls and IDPS devices). Such appliances are supported by most of the cloud providers and hence the companies may simply have to invest in the BI framework, or buy SaaS subscriptions. The performance of both arrangements is expected to be the same.

d) Perimeterised and deperimeterised model: In perimeterised, a company may like to isolate the server arrays holding business critical servers (like BI framework) and employ embedded distributed security components. This may be more expensive than deperimeterised setting in which no such boundaries are created. The performance of perimeterised arrays may degrade because the array sizing and provisioning of computing, networking and storage capacity may depend upon the paying power of

the company requesting for it. Deperimeterised model will be highly effective and will provide excellent returns.

e) Insourced or outsourced model: In the insourced model, a company may like to implement a self-hosted private cloud. Such a cloud may defeat the business advantages of having BI on the cloud and distributed security components embedded in the cloud. For example, such a cloud will have lesser elasticity and will be very expensive. Outsourced model will have excellent performance returns and will be cost effective because the cost of elasticity will be shared among multiple clients.

Finally, it is believed that embedding BI specific security controls (security attributes in the metadata repositories, separate database objects holding monitoring and activity logs, and table level encryption of data and security metrics details) and the rest of the security controls (antimalware, anti-spam, IDPS, etc.) may overload an array significantly. The rate of increase of elasticity will need to be significantly higher. The conventional security components (like IDPS, antimalware, anti-spam, web services security, etc.) require separate administration procedures, and have their own databases, which are continuously updated over the Internet (Al-Aqrabi et al. 2012). Hence, administration of conventional security components and BI specific distributed and embedded security components in the same array may be very challenging. In addition, it may not be feasible to combine the accesses provided to administrators of conventional security components and BI security and data components. Hence, it is recommended that the conventional security components should be kept in a separate server array on the same cloud in the UTM approach. This approach will ensure that both areas of securing BI on the cloud will be managed by different agents and there will be lesser chances of breaches in the data mart and warehouse data-files, and the OLAP cubes. The embedded security role is more of a DBA activity and hence this segregation is expected

to work. This will not affect performance provided both arrays are hosted on the same cloud served by a common cluster of network switches.

BI on the cloud is a large-scale array of database and OLAP application servers. The administration of such large-scale arrays will be carried out making use of XML document object models as data-files. The extraction, transformation and loading of data files will be carried using XML data-files and the DOM parsing feature of databases. The security controls of BI on cloud can be implemented by capturing source security attributes (called security metrics) defined in the source transactional databases and maintained in the metadata repositories. The metadata information will be exportable to XML format. The data transformation agents should record security related summaries and attach them with the multidimensional data hierarchies created and stored in data warehouses. In addition, an automatic relational database security tool should continuously monitor and log all changes to the data-files in separate relational database objects. This arrangement is called distributed and embedded security by the scholars those proposed them. In this research, two scenarios have been modelled in OPNET – (a) BI security using a UTM cloud (security-as-a-service), and (b) BI security using distributed and embedded components. The second scenario performed much better than the first scenario. With 3000 concurrent users, the first scenario returned unacceptable performance of DB queries and html page and object response times. There were significant number of TCP delays, TCP segment delays and retransmission counts. However, the first model performed very well and returned excellent performance on the cloud with 3000 concurrent users on the OLAP and database servers. Hence, it is evident that UTM cloud model may not work for BI security on the cloud. The BI security on cloud is further analysed based on the cloud-cube security model proposed by reference (Chea et al. 2011). Based on the simulation results, it is concluded that the external, open, deperimeterised and outsourced models are expected to return best results when employing distributed and embedded security

components to secure BI on the cloud. Finally, it is concluded that mixing BI specific security controls and general security controls on the same array may not be feasible administratively. Hence, it is recommended that both forms of security implementation may be split into two different server arrays on the same cloud. Such a model will keep the security administrators with the two roles separate, and will not mix the role of conventional security with security administration during data extraction, transformation and loading in a BI framework. In addition, it is expected that performance of BI transactions will not be affected if both server arrays are served by a common cluster of network switches and high-speed links.

4.6 **Summary**

In this chapter, a review of network modelling and simulation, introduction to OPNET toolkit, a review of network modelling in OPNET, and a description of the first, second, and third modelling scenario is presented. The three models described are related to cloud security, BI on the cloud, and BI security on the cloud scenarios. The BI security on the cloud is divided into two models – Model A and Model B. The modelling part helped in preparing the fundamentals for the three models to be created as the final outcome of this research. In this chapter, the simulation results of these models are presented. The simulation results of the four modelling scenarios have been presented and critically analysed. All the discussions presented in this chapter will be used as inputs to the finalised model, which is presented in the next chapter.

Chapter 5: Multiparty Authentication System (MAS)

5.1 **Introduction**

In this chapter, we present a multiparty authentication framework and the proposed model for securing BI on the cloud and we define five core protocols in our multiparty authentication system. The formal analysis enables us to re-examine assumptions used to develop our authentication protocols, to verify whether the objectives of our protocols are achieved through the intended actions. A comprehensive empirical study is performed to evaluate the scalability and the runtime overhead of the authentication system. Finally in this chapter, risks, ethics and legal implications are considered including mitigation factors and recommendations.

5.2 **Multiparty Authentication Framework in the Cloud**

In this section, the proposed framework for dynamic authentication interactions in a distributed environment is shown in Figure 31. The author propose adding a session authority cloud controlling sessions of multiple clouds. There shall be no concept of home or foreign clouds. Every cloud obeys the decisions made by the session authority cloud. The session authority cloud shall hold a large array of servers serving as a security vault. This vault holds authentication credentials and digital signatures of tenants of all clouds. The root keys of all the clouds are stored in the vault having folders identifying the clouds. An active tenant will "know" the root key of its own cloud. The security realms (sub-domains) are distributed among all the clouds and are identified through separate sub-domain keys. These keys will be stored in subfolders within the corresponding cloud folders. The concept of private key assignment against a digital signature will be adopted but only for entry into the relevant sub-domain of the cloud. In the multiparty session scenario, members of multiple sub-domains

may interact within a session. All such sessions will be identified by the session authority cloud. The session keys will comprise of root key of the cloud, sub-domain key, and the portion identifying the session. This means that there will be multiple session keys valid for a session, each having a common field for the session but varying fields for cloud root keys and sub-domain keys. There is no need for any negotiation among the clouds because the session authority cloud "knows" all the clouds and their sub-domains. The schematic of the proposed framework is shown in the below figure:

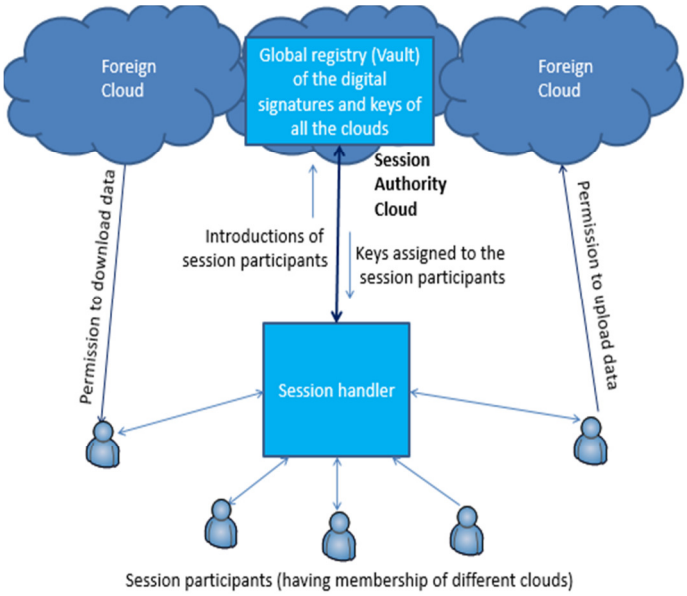

Figure 31: Proposed multiparty session authentication framework in cloud environment

The ground rules of the proposal are:

- ✓ Each session participant should be a tenant of at least one cloud in the multi-cloud framework controlled by the session authority cloud.
- ✓ If a potential participant is not a cloud member, the introducing participant will have to share credentials with it for joining its own cloud.

- ✓ Each session will have multiple keys valid. While the session key field will be common (refreshed on change of no of participants), the cloud root keys and sub-domain (security realm) keys will vary depending upon the membership profiles of the participants.

5.3 **Multiparty Authentication System for Securing BI on the Cloud**

The proposed system is shown in Figure 32, specifically addressed to scenarios of BI applications access on the clouds for dynamic secure interactions when members of different security realms want to access distributed BI services through a trusted principal. These scenarios are applicable when there are no direct authentication relationships between users and BI services in multiple Cloud systems located in different security realms.

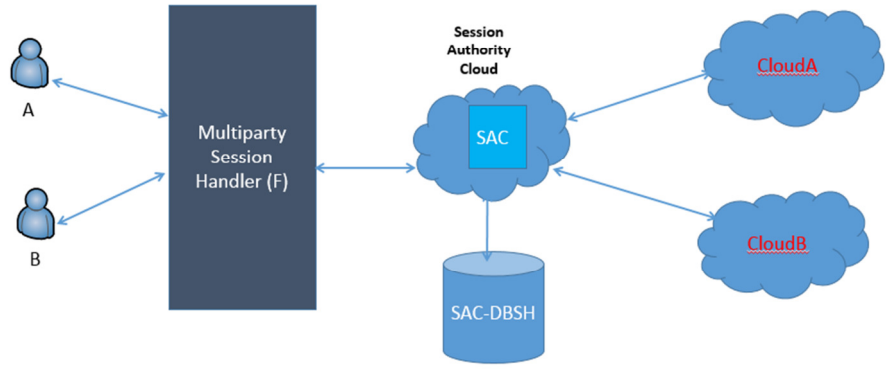

Figure 32: Multi-party authentication system for securing BI on the cloud

A – This is the principal that forwards the session request for a user (whom he/she trusts) of any security realm willing to access BI databases hosted on remote clouds (clouds A and B). The principal is the only one trusted by the session authority Cloud.

F – This actor is playing the role of multiparty session handler that collects the keys from the session requester and forwards to the Session Authority Cloud (SAC).

SAC – This actor is playing the role of a session authority cloud that authorizes the sessions based on matching of keys forwarded by F. It serves as the cloud certification authority (CCA).

SAC-DB – This actor is a supporting database that advises the SAC about matches/mismatches of keys sent by F.

Clouds A and B – These are clouds having membership with the session authority cloud for allowing access to their hosted resources from users authorized by the SAC.

The operation of the algorithm is described as the following:

The session begins with a user having membership in any security realm (Cloud C) that the trusted principal recognizes. We assumed to provide access to the BI database objects in clouds A and B if the SAC approves the request forwarded by the principal. It is also assumed that SAC will not entertain any request not forwarded by the principal. The user requesting access is neither a member of Cloud A nor a member of Cloud B. In essence, the user is a member of a security realm that is a different cloud (Cloud C), which is trusted by the SAC (which means that the third cloud is a member of the SAC-DB). Most importantly, the principal should recognize who is the user because the SAC trusts the principal for accepting the session request. Hence, the only way the user can gain access to BI database files on clouds A and B is to send a request to the session authority cloud through the session handler F. The session handler will only forward requests of the trusted principal and hence the requests need to be forwarded through the login of the trusted principal. The only resources the requesting user has are a root key of cloud C (by default available to all the users of Cloud C) and a sub-domain (security realm) key. The principal “A” places a request to the session handler to gain access to resources A and B for the user (the user knows their URLs but do not have any access to them).

On the request of the session handler (F), the principal A shares the root and sub domain keys of the user. These keys may be viewed as two packets of a finite size (example, 1024 Bytes each). F is just an intermediate system having no decision-making powers. Its role is to package the keys and forward to the SAC. The SAC checks the keys with the help of a database SAC-DB assisting it (may be viewed as a huge security vault having all root and sub-domain keys of the clouds registered with it). In essence, SAC-DB is a centralized registrar in the proposed algorithm.

On confirmation from SAC-DB, the SAC is approves access to BI database files A and B stored on clouds A and B respectively. It forwards its approval to the SAC's session handler (SAC-SH). The SAC-SH may be viewed as a separate dynamic database that caches all approvals from the SAC and forwards them to respective clouds for opening the accesses.

In this way, the SAC can be freed from this responsibility and allowed to focus on approving/rejecting sessions after verifying the keys records in the SAC-DB. It also is responsible for creating a key called the session key and sends to the SAC-SH along with the requested URLs (cloud BI database files A and B) and the ID of the requesting session handler. It may be noted that SAC-SH needs to know the identification of the requesting session handler because there may be a number of them interacting with the SAC. The session key may be viewed as an approval signature by the SAC.

The SAC-SH interacts with the clouds A and B, shares the SAC's approval (the session key), and requests for opening access to the resources. After verifying the approval of the SAC, the clouds A and B open the access to the BI database files for the session identified by the session key issued by the SAC. Thereafter, the SAC-SH sends confirmation to the requesting session handler (F). The authorized session key now becomes a combination of the cloud root key, the sub-domain (security realm) key, and the session key issued by the SAC. The session key will remain valid till the session is kept active by the trusted principal. Given

that multiple users may join the session through the trusted principal, same session can have more than one valid key (root key, security realm sub-domain key and the session key issued by the SAC). This key combination is unique for each user joining the session. If a user exits the session, the validity of the key combination is dropped such that he/she cannot re-enter the session using the same key combination. The user will need to request the trusted principal for re-entry and the whole process will begin again. For re-entry, the requesting user will get a new key combination again valid until his/her next exit from the session.

5.4 **Experiments**

In this section, a set of experiments are implemented on OPNET Modeller and Eclipse respectively to develop our multiparty authentication model and to test our multiparty authentication protocols efficiently and increase the reliability as well as performance.

5.4.1 Experiment on OPNET Modeller

In this section, we present a multiparty authentication system model for securing BI on on the cloud. The proposed system is specifically addressed to scenarios of BI applications access on the clouds when members of different security realms want to access distributed BI services through a trusted principal. These scenarios are applicable when there are no direct authentication relationships between the people of different security realms and the distributed BI services in multiple cloud system.

The entire multi-cloud model is created on OPNET modeller using HP9000 Superdome 64 CPU mega modular servers. These servers are the most powerful systems available in OPNET's model library in the academic edition. Each server can host hundreds of virtual machines. The model is shown in Figure 33. The object "A" in this model comprises an Internet cloud 1000 users, each representing a trusted principal by the SAC requesting a session each for a foreign user (member of a different security realm) on his/her

behalf. The objects F, SAC, SAC-DB, and SAC-SH are independent HP9000 Superdome servers whereas clouds A and B are collections of four HP9000 Superdome servers each.

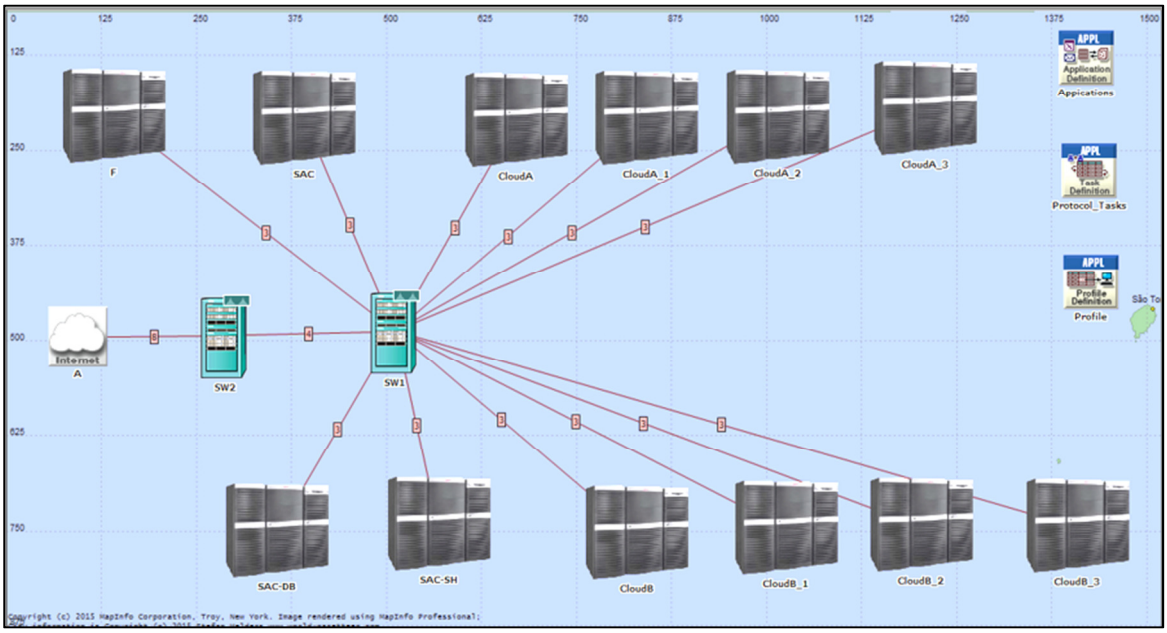

Figure 33: Multi-party authentication model

SW1 is an Internet switch connecting the trusted principals to the entire inter-cloud framework. SW2 is an internal switch of the inter-cloud framework interconnecting the clouds A and B with the core cloud certification authority (SAC in this model) and all its supporting services (F, SAC-DB, and SAC-SH). SW1 and SW2 are advanced Cisco chassis-based switches. The red lines represent 1000BaseX links and the numbers at the centre of each line represent the number of links per connection. For example, the red line connecting A with F has eight 1000BaseX links. Overall, this is quite a powerful network possible in academic edition of OPNET modeller with little scope of link and node level congestions for 1000 users connecting the network. Hence, whatever delays are found in the simulation results are because of the execution times of the phases of the authentication algorithm.

The OPNET tasks object has been used to define the phases of the authentication algorithm. Tables 4 and 5 illustrate how they have been modelled in OPNET modeller. Table

4 illustrates the phases configured without any phase-wise timeouts and Table 5 illustrates a timeout of 60 seconds per phase. Both the configurations are simulated separately in OPNET modeler. The phases are sequential, each considered as a request or a response between the stated nodes. In each phase, an appropriate amount of data is transferred as may be needed during practical operation of the phase. For example, the first phase A > F and the second phase F > A are requesting phases in which, the data transmission size is configured as 1024 bytes. However, the third phase A > F is a responding phase (A submits IDr and IDs to F) and hence the data transmission size is 4096 bytes.

Table 4: The algorithm steps are configured as individual protocol tasks in OPNET tasks creator object with no timeout

Phase Name	Start Phase After	Source	Destination	REQ/RESP Pattern	End Phase When	Timeout Properties	Transport Connection
A>F:Secure (Request, R1, R2)	Application Starts	A	F	REQ>RESP>	Final Response	Not Used	Default
F>A:Secure (Request, IDr, IDs)	Previous Phase Ends	F	A	REQ>RESP>	Final Response	Not Used	Default
A>F:Secure (Response, IDr, IDs)	Previous Phase Ends	A	F	REQ>RESP>	Final Response	Not Used	Default
F>SAC:Fetch(R1,R2):IF Valid(IDr, IDs)	Previous Phase Ends	F	SAC	REQ>RESP>	Final Response	Not Used	Default
F>SAC-DB:Verfiy(IDr, IDs)	Previous Phase Ends	SAC	SAC-DB	REQ>RESP>	Final Response	Not Used	Default
SAC-DB>SAC:Valid(IDr, IDs)	Previous Phase Ends	SAC-DB	SAC	REQ>RESP>	Final Response	Not Used	Default
SAC>SAC-SH:Invoke(Key,IDsess):Fetch (RI,R2)	Previous Phase Ends	SAC	SAC-SH	REQ>RESP>	Final Response	Not Used	Default
SAC:SH>CloudA:Secure(Request,R1): IF key(IDsess)	Previous Phase Ends	SAC-SH	CloudA	REQ>RESP>	Final Response	Not Used	Default
CloudA>SAC:SH:Secure(Access, R1)	Previous Phase Ends	CloudA	SAC-SH	REQ>RESP>	Final Response	Not Used	Default
SAC:SH>CloudB:Secure(Request, R2):IF key(IDsess)	Previous Phase Ends	SAC-SH	CloudB	REQ>RESP>	Final Response	Not Used	Default
CloudB>SAC:SH:Secure(Access, R2)	Previous Phase Ends	CloudB	SAC-SH	REQ>RESP>	Final Response	Not Used	Default
SAC:SH>F:Secure(Acess,R1,R2):Key (IDsess)	Previous Phase Ends	SAC-SH	F	REQ>RESP>	Final Response	Not Used	Default

Table 5: Timeout introduced in each phase of the authentication protocol

Phase Name	Start Phase After	Source	Destination	REQ/RESP Pattern	End Phase When	Timeout Properties	Transport Connection
A>F:Secure (Request, R1, R2)	Application Starts	A	F	REQ>RESP>	Final Response	Used	Default
F>A:Secure (Request, IDr, IDs)	Previous Phase Ends	F	A	REQ>RESP>	Final Response	Used	Default
A>F:Secure (Response, IDr, IDs)	Previous Phase Ends	A	F	REQ>RESP>	Final Response	Used	Default
F>SAC:Fetch(R1,R2):IF Valid(IDr, IDs)	Previous Phase Ends	F	SAC	REQ>RESP>	Final Response	Used	Default
F>SAC-DB:Verfiy(IDr, IDs)	Previous Phase Ends	SAC	SAC-DB	REQ>RESP>	Final Response	Used	Default
SAC-DB>SAC:Valid(IDr, IDs)	Previous Phase Ends	SAC-DB	SAC	REQ>RESP>	Final Response	Used	Default
SAC>SAC-SH:Invoke(Key,IDsess):Fetch (RI,R2)	Previous Phase Ends	SAC	SAC-SH	REQ>RESP>	Final Response	Used	Default
SAC:SH>CloudA:Secure(Request,R1): IF key(IDsess)	Previous Phase Ends	SAC-SH	CloudA	REQ>RESP>	Final Response	Used	Default
CloudA>SAC:SH:Secure(Access, R1)	Previous Phase Ends	CloudA	SAC-SH	REQ>RESP>	Final Response	Used	Default
SAC:SH>CloudB:Secure(Request, R2):IF key(IDsess)	Previous Phase Ends	SAC-SH	CloudB	REQ>RESP>	Final Response	Used	Default
CloudB>SAC:SH:Secure(Access, R2)	Previous Phase Ends	CloudB	SAC-SH	REQ>RESP>	Final Response	Used	Default
SAC:SH>F:Secure(Acess,R1,R2):Key (IDsess)	Previous Phase Ends	SAC-SH	F	REQ>RESP>	Final Response	Used	Default
F>A:Secure(Acess,R1,R2):Key(IDsess)	Previous Phase Ends	F	A	REQ>RESP>	Final Response	Used	Default

The data sizes have been configured accordingly in all the phases of the algorithm. A subsequent phase does not begin unless the previous phase has ended. Thus, if a phase fails to complete, the subsequent phase will not begin at all. A phase will be deemed as ended only when a final response has arrived from the requested node to the requesting node. Thus, if there is congestion on the network and a timeout has been configured for each phase, the session will be dropped if any of the subsequent phases fails to execute successfully.

(a) The phases have been packaged in an application called as “Protocol_Tasks” in the model. As shown in Table 10 (Appendix B), Protocol_Tasks is a custom application, which in turn is designed using the tasks object in OPNET modeller. One needs to enter the attributes in this field and select the task object packaged with all the phases configured.

(b) The database application has been configured for running on SAC-DB only. It is a OPNET's default high load task format. There was no need to configure it manually because it is not the focus of this research.

(c) The applications are executed using the profiles object, as shown in the following table. Protocol_tasks and SAC-DB have been configured to execute independent of each other such that they do not cause a conflict during simulation. Both applications have been assigned a start offset of 5 to 10 seconds. However, the start offset of the network itself has been configured at 100 to 110 seconds. This is because the network is large and should be given enough time for completing the tasks of the routing protocol. In this model, the routing protocol selected is RIPv2 (which is OPNET's default).

Table 6: Configuring the profiles object for executing the applications

Attribute	Value
Name	Profile
Profile Configuration	
- Number of Rows	1
Protocol_Tasks	
- Profile Name	Protocol_Tasks
Applications	
- Number of Rows	2
Protocol_Tasks	
- Name	Protocol_Tasks
- Start Time Offset (seconds)	Uniform (5,10)
- Duration(seconds)	End of Profile
Repeatability	
SAC-DB	
- Name	SAC-DB
- Start Time (seconds)	Uniform (5,10)
- Duration (seconds)	End of Profile
Repeatability	
- Operation Mode	Simultaneous
- Start Time (seconds)	Uniform (100,110)
- Duration (seconds)	End of Simulation
Repeatability	Once at Start Time

Before explaining the simulations, it is essential to clarify how the nodes recognize whom to contact for executing a phase. This clarification is needed to map the symbols (A, F, SAC, SAC-DB, Cloud A, and Cloud B) with actual servers (names of servers in the network) on the network. For simplicity, the symbols and server names have been kept the same in the model. However, in actual clouds they will be completely different. The configuration needed for this mapping in OPNET modeller is called destination preferences as shown in Table 11, and 12 (Appendix B). This is a complex configuration in OPNET that needs to be configured carefully by matching the source-destination relationships with the phases of the customer application (Protocol_Tasks authentication algorithm). In destination preferences, a node is allowed to communicate with only those nodes that it is supposed to interact for executing the algorithm. Hence, A is allowed to communicate only with F and F is allowed to communicate only with A and SAC.

It may also be noted that a self-relationship (A communicating with A and F communicating with F) is established in destination preferences, as well. This is done to ensure that the node is able to communicate with itself whenever required by OPNET.

Finally, the attachment of a profile with a node is important to instruct it about what it needs to execute. This is done by identifying the profile within the node configuration. In this model, all nodes are configured to execute the "Protocol_Tasks" whereas SAC-DB is configured to execute "Protocol_Tasks" as well as the database application.

5.4.2 Experiment on Eclipse

With the help of Eclipse, we created our five multiparty authentication protocols in our multiparty authentication system. It is used to develop the prototype authentication system based on the proposed framework. The experimental configuration is shown in (Appendix A).

Figure 34 illustrates the main screen of the session authority cloud program. The program set consists of 3 parts:

- SAC: Session Authority Cloud program
- Cloud: User program
- Service: service program

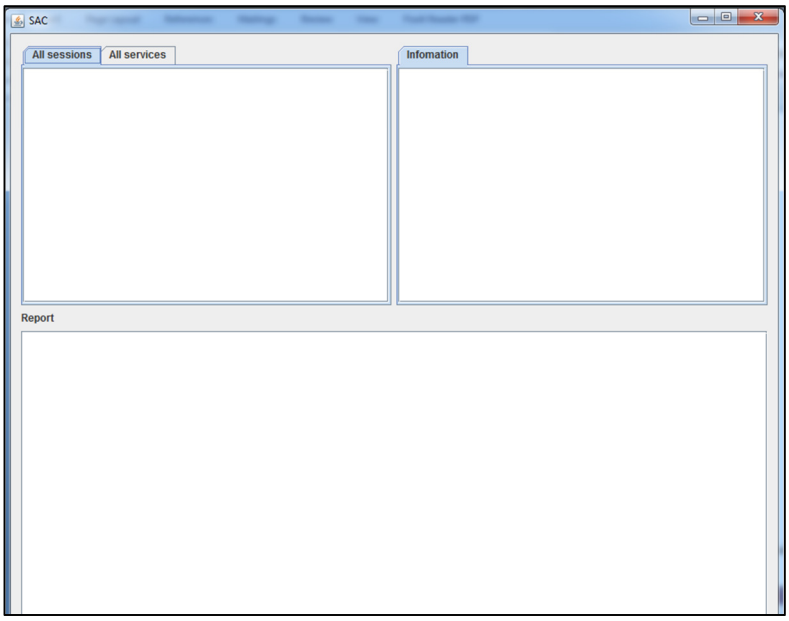

Figure 34: SAC program main screen

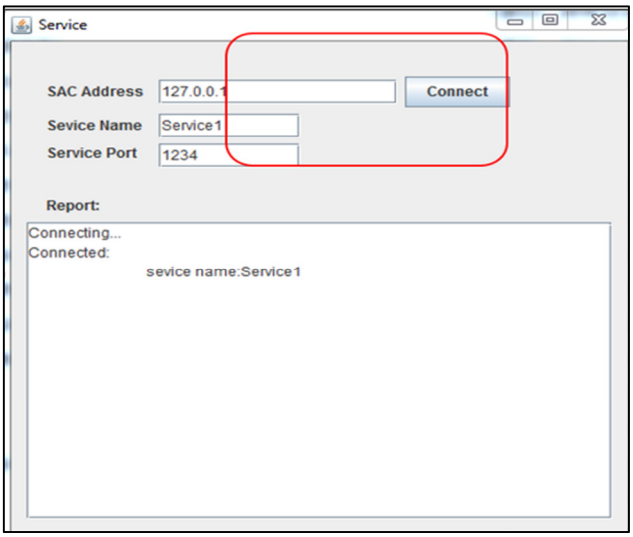

Figure 35: Cloud user connecting to service

Figure 35 presents the cloud user connecting to service. Figure 36 illustrates use case a multiparty authentication system on the cloud. This Multiparty authentication system can he lp session users authenticate their session memberships to simplify the authentication process es within multiparty sessions.

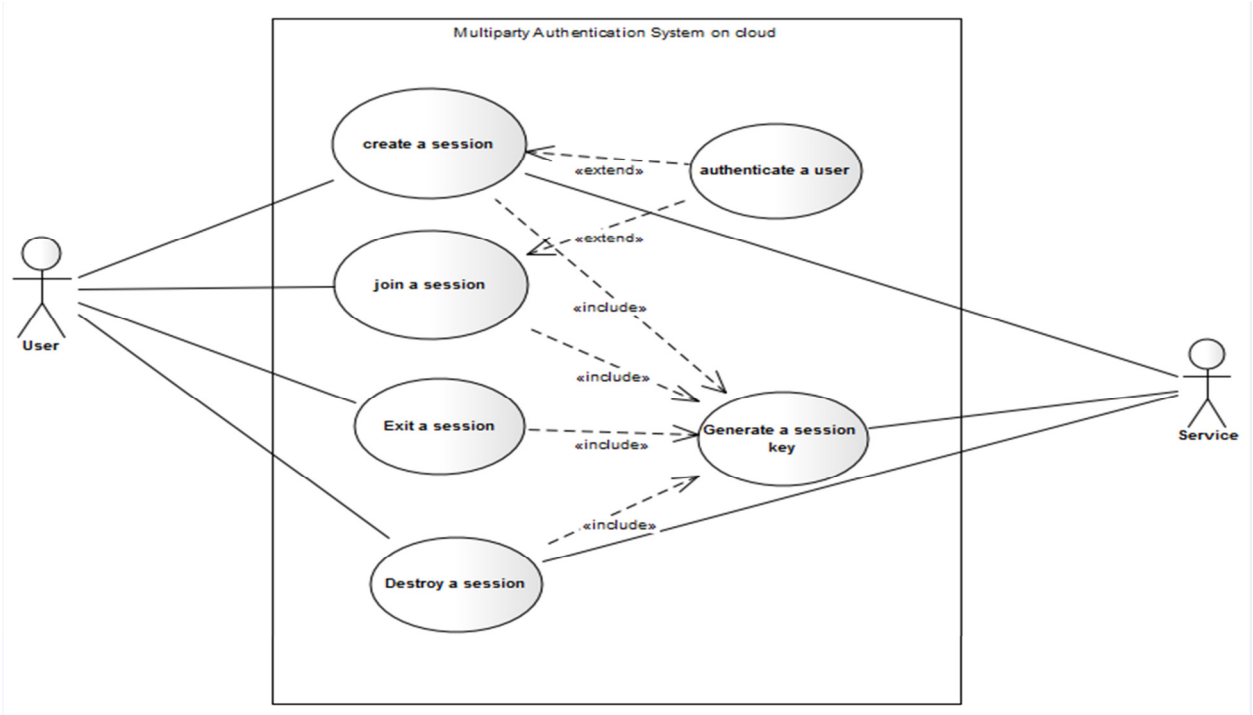

Figure 36: Use case Multiparty authentication system diagram

The following UML class diagrams of the proposed system:

- SAC

This program represents SAC program.

- Session Handler

This class is one for session management. This class is responsible for session approval (creating a session and permitting to access resources) and deleting sessions.

- Node

This class is one for communicating with session users. Whenever a user creates or joins a session, an instance of this class is created and registered in node list and associated

with the corresponding session. This class is responsible for processing requests of its corresponding user.

- Session

This class represents a multipart session. This class is responsible for management of users that belong to it and management session key. Whenever a session is created, Session Hander creates an instance of this class.

- Service

This class is one for communicating with services. This is responsible for forwarding requests of users to service and forwarding responses of services to users through Node. When a session is created, an instance of this class is created and registered in the session and associated with the corresponding service against each service that the creator of the session.

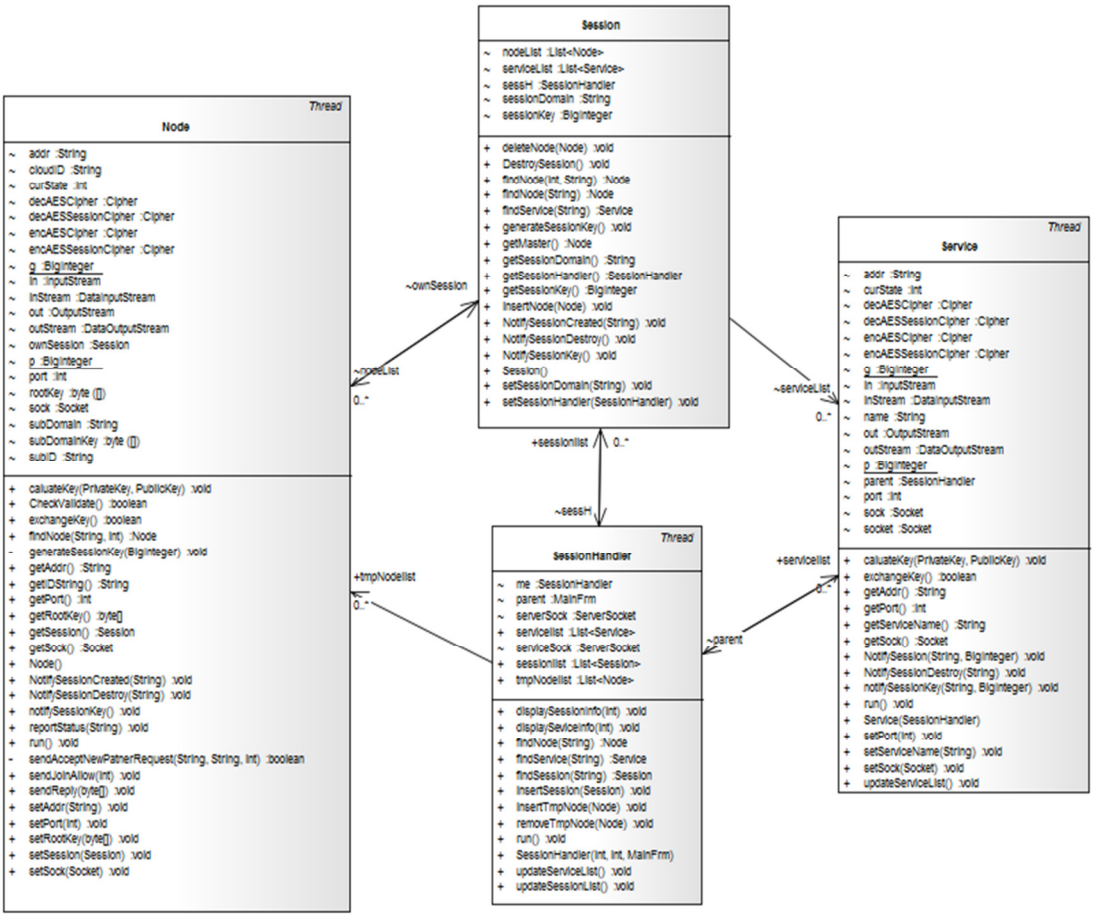

Figure 37: Classes of session authority cloud

- Cloud

This program represents a Cloud program.

- UserInterface

This class plays the role of user interface and a store of information about SAC. A user can view session list and service list and send requests of creating or joining a session.

- Agent

This class is responsible for communicating with SAC.

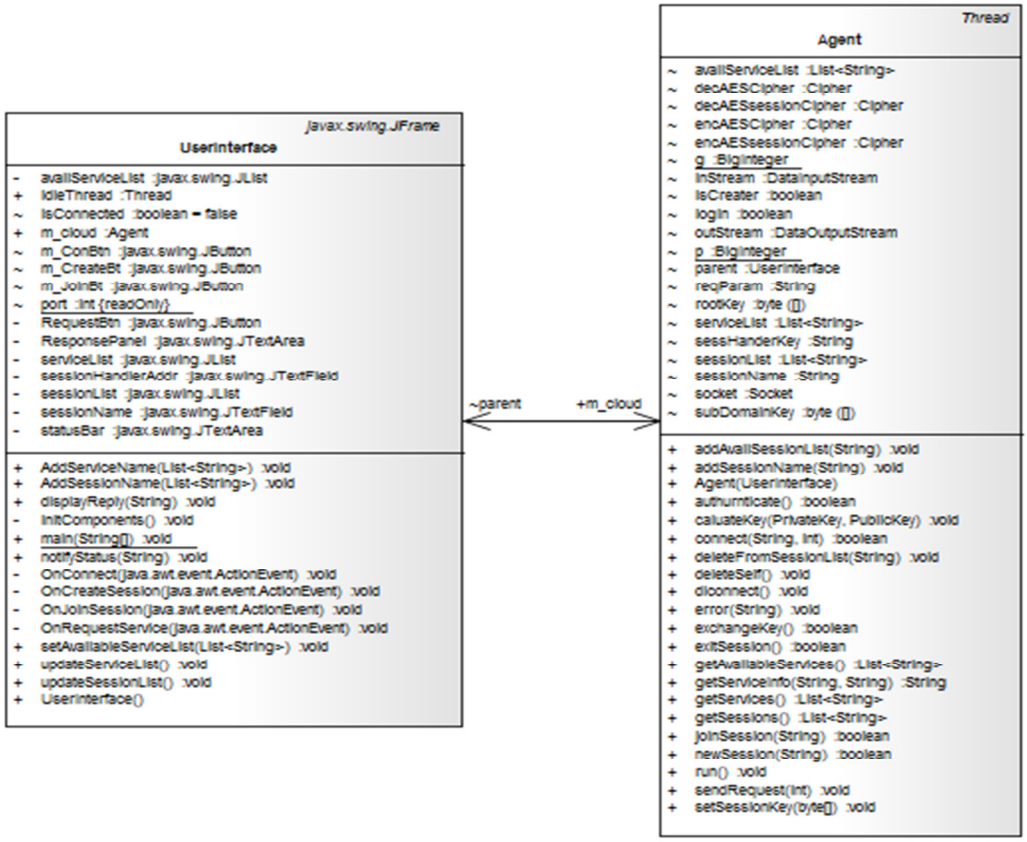

Figure 38: Cloud class diagram

- Service

This program represents a service. This only has very simple function.

- Service

This class is to let a person observe status of service (user call or login to session).

- Service Thread

This class is for processing requests of users. This class is for storing session information (session key, session identifier). Each instance of service Thread has one instance of this class.

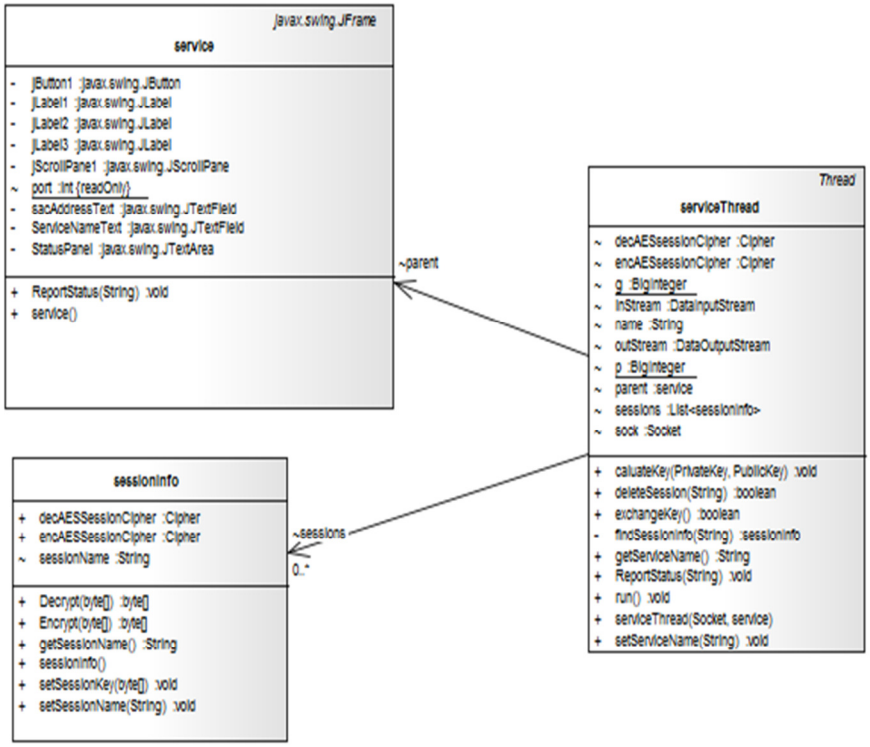

Figure 39: Service class diagram

5.4.2.1 A Scenario of a Business Session

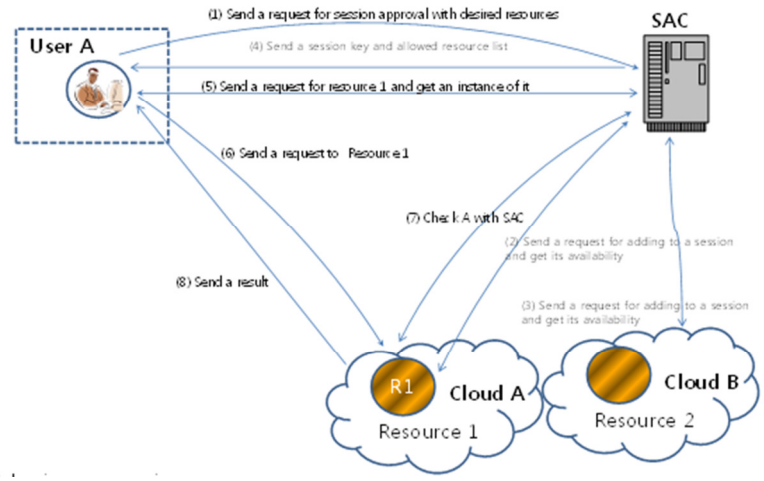

Figure 40: A Business Scenario

As illustrated in Figure 40, In this scenario, first user A contacts SAC to start a new session for his/her work, and forwards session name(ID) and desired resources list in step(1). SAC then verifies A' identity and creates a new session instance to control the new business session. SAC also generates the key of the new session and then registers the session its session list. Also SAC registers A in the session. Next, SAC invokes the resources A wants, so checks if available (step (2)(3)). After receiving a reply from resources, SAC then sends a response of session approval with session key and available resources list (step (4)). Before calling Resource1, A sends a request for the Resource1 to SAC. SAC then contacts the Resource1. Resource1 invokes a new instance R1 and send back its identifier to SAC. SAC records R1 in the session and sends it to A (step (5)). Next, after receiving a response from SAC, A sends a request to R1 (step (6)). Then R1 first contacts SAC to check whether A is a valid session user in step (7). Once it is validated, R1 sends a response of results that A requires (step (8)).

5.5 **Multi-Party Authentication Protocols**

In this section, we present five core protocols in our multiparty authentication system using the well-known Logic of Authentication or (BAN logic). Protocol 1 performs authentication for session approval, Protocol 2 for adding a new user to an existing session, Protocol 3 for accepting a new session User, Protocol 4 for leaving a session, and Protocol 5 for ending a session.

5.5.1 Diffie-Hellman Algorithm

The Diffie-Hellman (D-H) algorithm is a public key algorithm. D-H introduced by Diffie and Hellman in 1976, was the one of the earliest practical examples of public key exchange implemented within the field of cryptography. This algorithm allows two users that have no prior knowledge of each other to establish a shared secret key over an insecure channel. In the Multiparty authentication protocols, the Diffie-Hellman algorithm is employed to help session user secure their communication. Hada and Maruyama (2002) demonstrate the need for multi-party session authentication protocol, if a multi-party session is constructed out of multiple two-party sessions, it is very hard in some cases for a session user to determine and verify whether the service instance it contacts is a member of the same multi-party session. In a cross-realm authentication, the techniques used in two-party session does not address such Heterogeneous Cross-Realm Authentication issues, which requires credential conversion and the establishment of authentication paths.

5.5.2 BAN Notations

BAN is a logic of belief (Abadi and Needham 2003). BAN logic was introduced by Burrows, Abadi and Needham. BAN logic is used to analyse authentication protocols by deriving the beliefs that honest principals in protocol. BAN consists of three stages to analyse any protocol. The first stage is to express the initial assumptions, and goals as statement to

translate them to symbolic notations. The second stage is to verify the goal whether the goals are in fact reached. Lastly, a group of rules are performed to acquire the authentication goal.

Assume there are three principals, Alice (A) and Bob (B), Server (S) for example, A might come to believe that a key (^{K}AS) she has received from a (S) is a good key (^{K}AB) for a communication session with (B). To idealise the protocol by replacing concrete messages by idealised messages in the protocol into logical formulae. For example, if a (S) sends (A) a session key (^{K}AS) inside an encrypted message, the key (K) might be replaced with logical formula that means that the key (K) is good. We could then apply the inferences rules based on (A)'s ability to decrypt the Key (K) and other assumptions that would also lead to the conclusion that (A) believes that the received key (^{K}AB) is good for talking with (B).

Before the proofs of correctness of our authentication protocol algorithms, let us state the following notation for formal definitions and proofs.

A, B	session users
RA, RB	resources on the cloud
SAC	session authority cloud
ID_A	identifier of A
S	multiparty session with identifier ID_S
Pri(A)	private key of user A
Pub(B)	public key of user A
X, Y	statements
[X, Y]	composite message composed of messages X and Y, i.e.,$[ID_{SAC},\ ID_A,\ K_S,\ ID_S,\ \mathrm{N}+1]$
↑Pub(A)	Pub(A) is good i.e., its corresponding Pri(A) will never be discovered by any other users and Pub(A) is not weak
#N	N is fresh, i.e., M has not been sent in a message at any time before the current run of the protocol.
SP(A, S)	statement that A is a session user of S. Particularly, SP(SA, S) is always true.

$A \overset{K^{-}_{(A,B)}}{\longleftrightarrow} B$ $K_{(A,B)}$is A's secret key to be shared with B. No third user aside from A and B can deduce $K_{(A,B)}$, but A has not yet get confirmation from B that B knows $K_{(A,B)}$.

$A \overset{K^{+}_{(A,B)}}{\longleftrightarrow} B$ $K_{(A,B)}$is is a key held by A. No third user aside from A and B can deduce $K_{(A,B)}$ and A has received key confirmation from B which indicates that B actually knows$K_{(A,B)}$

A|≡ X A believes that statement X is true.

A|⟹ X A is an authority on X, i.e., A has jurisdiction over X.

P sees X. P has received some message X and is capable of reading and repeating it.

$[X]_K$ This represents the formula X encrypted under the key K.

$\overset{Pub(A)}{\longrightarrow}$A A has Pub(A) as a public key. The matching secret key will never be discovered by any user except A or a user trusted by A.

C(A) Certification of A. It includes the cloud root key + subdomain key and signature. It is to be sent to SAC after encrypted under A's private key Pri(A).

Ks, Ks' the session key of the session S. Ks is the old, Ks' is the new.

Exit(A, S): statement that A want to exit from session S.

Rm(S): statement that SAC removed session S.

5.5.3 Rules of Inference

We use the extended BAN logic to analyse formally the correctness for the Protocols. We first introduce some deduction rules to be used by our correctness proofs. These rules are specified in (Abadi and Needham 2003).

BAN Rules:

Rule 1. A|≡ (X, Y) ⟹ A|≡ X and A|≡ Y,

A sees (X,Y) ⟹ A sees X

A believes a set of statements if and only if A believes every individual statement, respectively.

Rule 2. A|≡ #M ⟹ A|≡ #(M, N) and A|≡ #(N, M)

The whole message is believed to be fresh if a part of a message is believed to be fresh.

Rule 3. $A|\equiv B| \Rightarrow X,\ A|\equiv B\ |\equiv X \Rightarrow A|\equiv X$

This inference rule states that if A believes that B has a control over statement X, and if A believes that B believes X, then A should believe X.

Rule 4. $A|\equiv\uparrow Pub(A),\ A|\equiv\uparrow Pub(B) \Rightarrow A \overset{K^{-}_{(A,B)}}{\longleftrightarrow} B$

In this rule, if A has B's public key and believes that the public keys of A and B are both good, A can believe that the secret is shared with no party other than B although unconfirmed.

Rule 5. $A|\equiv A \overset{K^{-}_{(A,B)}}{\longleftrightarrow} B$, A sees $[X]_{K(A,B)}$, $A|\equiv \#X, \Rightarrow A|\equiv A \overset{K^{+}_{(A,B)}}{\longleftrightarrow} B$

If A believe the secret $K_{(A,B)}$ is shared with no party other than B(but A does not know B knows) and X encrypted by the secret is fresh, then A can believe that the secret is confirmed by B.

Rule 6. $A|\equiv A \overset{K^{+}_{(A,B)}}{\longleftrightarrow} B$, A sees X, $A|\equiv \#X, \Rightarrow A|\equiv B| \equiv X$

If A believes that the secret is shared with no party other than B and is confirmed, and X is fresh, then A can believe that B believes X.

Rule 7. $A|\equiv B \overset{K}{\leftrightarrow} A$, A sees $[X]_K \Rightarrow$ A sees X,

If A knows the secret shared with B and see X encrypted by the secret, then can see X.

$A|\equiv \xrightarrow{Pub(B)} B$, A sees $[X]_{Pri(B)} \Rightarrow$ A sees X.

If A knows the public Key of B and sees X encrypted by B's private Key, then A can see X.

5.6 **Correctness Proofs for the Protocols**

To prove the protocols using BAN Logic, it has to be more briefly, but its idea and functions MUST NOT be modified. To do so, Multiparty Session Handler, SAC, Vault and SAC Handler can be considered making into one entity (SAC), because most authentication functions are concentrated upon SAC and the messages can be merged suitably without modified their meanings. Therefore, we can simplify the system as shown in Figure 39.

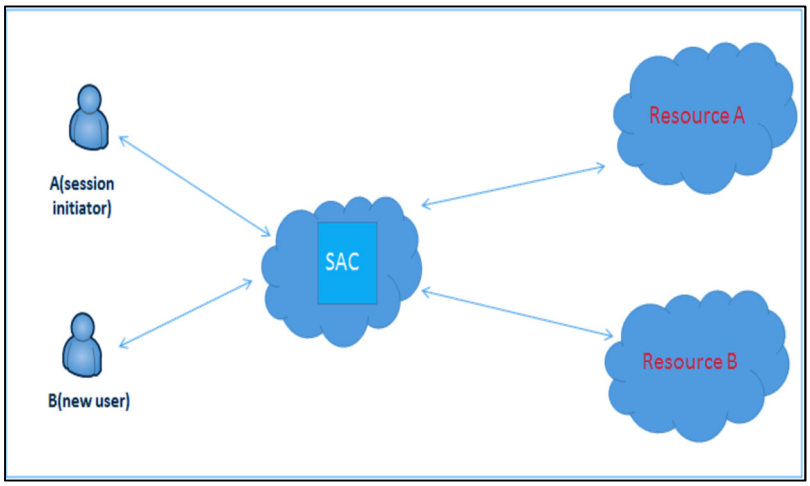

Figure 41: Simplified system for BAN Logic

To prove the correctness of the security protocol. It is important to show whether a protocol indeed achieves its security goals after running the protocol under the stated assumptions. Therefore the following theorems were proposed.

5.6.1 Protocol 1: Session Approval

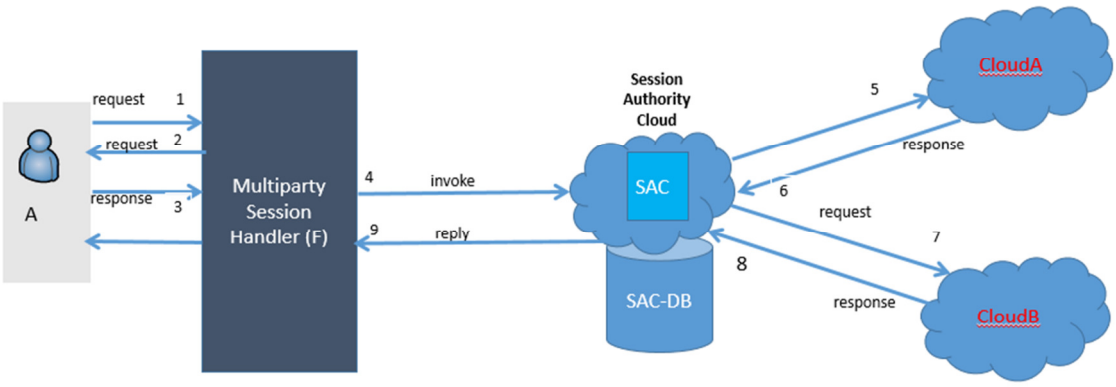

Figure 42: Session approval protocol

Figure 42 illustrates the session approval protocol. User A sends a request to create a new session in order to access the BI application on Cloud A or B. F sends a request for User A's keys (certificate). N1 is a nonce for preventing the replay attack and for matching the

request and reply. User A sends his/her certificate, which contains a root key and subdomain key, and the certificate is encrypted with User A's private key. Here, C(A) consists of IDr and IDs. Pri(A) is User A's private key. For the Digital Signature, C(A) is encrypted by Pri(A). F generates a new session ID and sends it, along with User A's request, to SAC in order to verify User A's identity and to approve a new session. SAC-DB uses User A's public key, which is stored in the vault, to decrypt the encrypted C(A). Then it can verify User A's identity by checking IDr and IDs. If User A's identity is valid, then SAC generates a session key and sends a request to Cloud A to access the BI application. SAC then requests access and notifies IDsess and its key. N2 is a nonce. Cloud A stores the session ID and key in its registry or cache and then sends a response to SAC. SAC sends a reply for session approval to F. Then, F sends a response for session approval to access the BI application on Cloud A or B.

The details of the messages transported in this protocol are presented as follows, where "A->B" means that A sends a message to B.

① A → SAC: $\left[ID_A,\ ID_{SAC},\ ID_R,\ [C(A)]_{Pri(A)},\ ID_S,\ N_1\right]_{K_{(A,\ SAC)}}$

② SAC → RA: $[ID_{SAC},\ ID_{RA},\ ID_S,\ K_S,\ N_2]_{K_{(SAC,\ RA)}}$

③ RA → SAC: $[ID_{RA},\ ID_{SAC},\ N_2 + 1]_{K_{(SAC,\ RA)}}$

④ SAC → A: $[ID_{SAC},\ ID_A,\ K_S,\ ID_S,\ \mathrm{N} + 1]_{K_{(SAC,\ A)}}$

When the user A tries to create a new session including Resource A (RA), A first sends a request to SAC (message (1)). This request consists of the credential of A, the name of Resource A and the session name to create. *(Indeed, in implementation the message (1) is sent by dividing to 2 messages. The first message includes the credential of A. after*

authenticating against the credential is success, the second which includes the names of session and service (resource) is sent. But these messages are merged to one by formalising and simplification for BAN Logic.) SAC then creates a new session and informs RA that RA has become a member of a new session (message (2)). After receiving a confirmation from RA (message (3)), the SAC sends back a response of session approval with session key (message (4)). All the messages are protected by the secret keys generated with D-H algorithm.

The Security goals of the protocol for session approval include the following:

1) verifying the identity A who are going to create a new session

2) building a session key to be shared among session members(RA, RB)and SAC

So,the security goals can be formally described as follows:

SAC sees C(A) ,that is, SAC must verify that A possess keys and verify signature.

A|≡Ks , RA |≡Ks, RA|≡Ks, this means session members must share the session key.

The assumptions of this protocol are formally described as follows:

$\mathbf{SAC}|\equiv\uparrow \mathbf{Pub}(\mathbf{SAC})$, $\mathbf{A}| \equiv\uparrow \mathbf{Pub}(\mathbf{SAC})$, $\mathbf{RA}| \equiv\uparrow \mathbf{Pub}(\mathbf{RA})$: users believe that their private keys are secure.

$\mathbf{A}|\equiv\uparrow \mathbf{Pub}(\mathbf{SAC})$, $\mathbf{SAC}| \equiv\uparrow \mathbf{Pub}(\mathbf{A})$, $\mathbf{SAC}|\equiv\uparrow \mathbf{Pub}(\mathbf{RA})$, $\mathbf{RA}| \equiv \mathbf{Pub}(\mathbf{SAC})$: they believe other's keys are secure.

$\mathbf{A}|\equiv \#\boldsymbol{N_1}$, $\boldsymbol{SAC}| \equiv \#\boldsymbol{N_1}$, $\boldsymbol{SAC}| \equiv \#\boldsymbol{N_2}$, $\boldsymbol{RA}| \equiv \#\boldsymbol{N_2}$**:** Users believe the nonce N_1,N_2 are fresh, that is, those have not been sent.

$\boldsymbol{A}| \equiv \mathbf{SAC}\vDash \mathbf{Ks}$: A believes SAC can control session key **Ks**
$\boldsymbol{RA}| \equiv \mathbf{SAC}\vDash\mathbf{Ks}$: RA believes SAC can control session key **Ks**

Theorem1.The goals of the protocol for session approval are satisfied under the assumptions of the protocol.

Proof. It is needed to deduce SAC sees C(A), A|≡ Ks , RA |≡Ks and RA|≡Ks from the assumptions of the protocol.

Firstly, let's achieve the first goal, SAC sees C(A).

From SAC |≡↑ Pub(SA) and SAC|≡↑ Pub(A), it follows that SAC |≡ SAC $\overset{K^{-}_{(A,\ SAC)}}{\longleftrightarrow}$ A by Rule 4.····· (i)

Formula (i) means SAC believe that the key shared with A , $K_{(A,\ SAC)}$ is secure, but SAC has not yet get the confirmation from A.

Then from the message①and SAC |≡♯ N, it yields that

SAC |≡♯ $\left[\mathrm{ID_A},\ \mathrm{ID_{SAC}},\ \mathrm{ID_{RA}},\ \mathrm{ID_{RB}}, [\mathrm{C(A)}]_{\mathrm{Pri_{(A)}}}, \mathrm{N_1}\right]$ by Rule 2.····· (ii)

Furthermore, from (i), (ii) and message ①, we can deduce that

SAC |≡ SAC $\overset{K^{+}_{(A,\ SAC)}}{\longleftrightarrow}$ A by Rule 5.····· (iii)

That is, SAC believes the key $\boldsymbol{K_{(A,\ SAC)}}$ is secure and SAC has get confirmation from A.

We can obtain SAC sees $[\mathrm{ID_B},\ \mathrm{ID_{SAC}},\ \mathrm{ID_S},\ [\mathrm{C(A)}]_{Pri(A)}, \mathrm{N_1}\ ]$ and

SAC sees $[\mathrm{C(A)}]_{\mathrm{Pri_{(A)}}}(\cdots iv)$ by the message①, (iii), Rule 7 and Rule1.

Consequently, we can obtain **SAC sees C(A)**by SAC|≡$\overset{Pub(A)}{\longrightarrow}$ A , (iv) and Rule7.

This means that SAC can see C(A) by decrypting it with A' public key and so *can verify the identity of A*.

Secondly, for second goal.

From A |≡ A $\overset{K^{+}_{(A,\ SAC)}}{\longleftrightarrow}$ SAC, A |≡♯ N, (iv) and Rule 6, we can deduce A |≡ SAC | ≡ $\mathrm{K_S}$:··· (v)

Then, from A |≡ SA ⊨ $\mathrm{K_S}$, (v) and Rule 3, we can obtain **A |≡ $\mathbf{K_S}$** .

That is, A can know the session key - **Ks**.

The remnants can also be deduced through a similar procedure.

Thus we can verify the session key is to be shared among the session members.

Hence the theorem.

So far we have proved the following.

- SAC can verify the identity of A exactly.
- All users of a session can share the secret key.

5.6.2 Protocol 2: Adding a User to an Existing Session

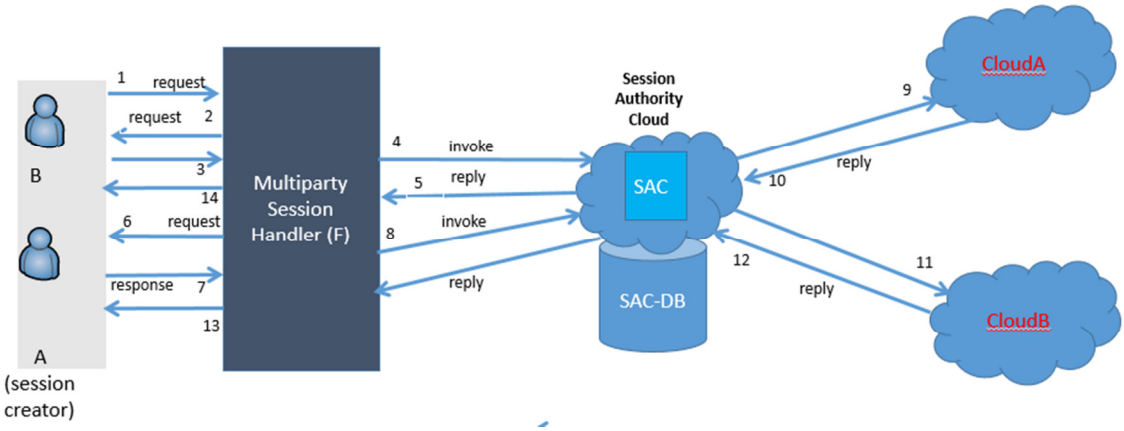

Figure 43: Adding a new user to an existing session

Figure 43 illustrates the protocol for adding a new user to an existing session. User B wants to join the session created by User A, IDsess. Steps 1–5 are identical to protocol 1. Thus, so far, SAC and F have verified User B's identity. F then asks if User A wants to allow User B to join. User A sends a response that either accepts or rejects the request. If User A accepts User B, then F sends a request to add User B to the session with IDsess and regenerates a new session key. SAC appends User B to the session, regenerates the session key, Ksess2, and then notifies all session members that the session key has been changed.

Cloud A stores the session ID and key in its registry or cache and then sends a response to SAC to access the BI application. SAC sends a reply for session to F, notifies User A of a new key and sends a response to User B to join the session.

The details of the messages transported in this protocol are presented as follows:

① **B→ SAC** $[ID_B,\ ID_{SAC},\ ID_S,\ [C(B)]_{Pri(B)},\ N_1]_{K_{(B,\ SAC)}}$

② **SAC→ A** $[ID_B,\ ID_S,\ ID_A,\ N_2]_{K_S}$

③ **A→ SAC** $[ID_A,\ ID_{SAC},\ ID_B,\ \mathbf{SP(B,S)},\ ID_S,\ N_2+1]_{K_S}$

④ **SAC→ A** $[K'_S,\ N_2+2\ ]_{K_S}$

⑤ **SAC→ B** $[K'_S,\ N_1+1]_{K_{(B,\ SAC)}}$

⑥ **SAC→ RA** $[ID_{SAC},\ ID_{RA}\ ,\ ID_S,\ K'_S,\ N_3]_{K_S}$

⑦ **RA→ SAC** $[\ N_3+1]_{K_S}$

When the user B join the session S (created by the user A), B first sends a request to SAC (message (1)). The SAC asks A (session creator) whether A wants to accept or not (message (2)). After receiving a confirmation from A (message (3)), SAC regenerates the session key and notifies creator A that the session key is updated(message(4)). Then SAC send a response of joining the session to B with the session key (message (5)). And then SAC informs all other session members of updating the session key (message (6)). The members send back confirmations.

The integrity of the message (1) (5) are protected by the secret key generated by D-H algorithm and the ones of messages (2), (3), (4), (6), (7) are protected by the secret key shared within the session S.

Security goals of the protocol for adding a user to an existing session include the following:

1) verifying the identity B who are going to join a session

2) accepting B as a new user

regenerating and distributing the session secret key to be shared among session members(RA, RB, B) and SAC

The security goals are formally described as follows:

SAC sees C(B) ,that is, SAC must verify that B possess keys and verify signature.

then, $\mathrm{SAC}| \equiv \mathrm{SP}(\mathrm{B},\mathrm{S})$ means accepting B as a new user.

$\boldsymbol{SAC}| \equiv \boldsymbol{K_s'}\ \boldsymbol{A}| \equiv \boldsymbol{K_s'}\ \boldsymbol{B}| \equiv \boldsymbol{K_s'}\ \boldsymbol{RA}| \equiv \boldsymbol{K_s'}\ \boldsymbol{RB}| \equiv \boldsymbol{K_s'},$ means session members must share the session key again.

The assumptions of this protocol are formally described as follows:

$\mathbf{A}| \equiv \mathbf{A} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC}$: A believes that SAC and A has the same session key and can use it to exchange of messages with SAC.

$\mathbf{B}| \equiv \uparrow \mathbf{Pub}(\mathbf{B})\ , \boldsymbol{SAC}| \equiv \uparrow \mathbf{Pub}(\mathbf{SAC}), \boldsymbol{B}| \equiv \uparrow \mathbf{Pub}(\mathbf{SAC}), \mathbf{SAC}| \equiv \uparrow \mathbf{Pub}(\mathbf{SAC})$:

users believe their private keys remain secure.

$\mathbf{SAC}| \equiv \mathbf{A}| \Rightarrow \mathbf{SP}(\mathbf{B},\mathbf{S})$: SAC believes A can control the B' joining to the session S.

$\boldsymbol{A}| \equiv \mathbf{SAC}\ | \Longrightarrow \boldsymbol{K_S'}$: A believes SAC can control session key Ks'.

$\boldsymbol{SAC}| \equiv \#\boldsymbol{N_1}\ ,\ \boldsymbol{SAC}| \equiv \#\boldsymbol{N_2}$, SAC believes the nonces $\mathrm{N}_1, \mathrm{N}_2$ are fresh.

$\boldsymbol{A}| \equiv \#\boldsymbol{N_2},\ \boldsymbol{B}| \equiv \#\boldsymbol{N_1},\ \boldsymbol{B}| \equiv \mathbf{SAC} \Mapsto$Ks': B believes SAC can control session key Ks'.

Theorem2. The goals of this protocol are satisfied under the assumptions of the protocol.

Proof. It is needed to deduce SAC sees C(A), A|≡ Ks , RA |≡Ks and RA|≡Ks from the assumptions of the protocol.

SAC |≡ SP(B, S), SAC | ≡ K_s', A |≡ K_s', B | ≡ K_s' , RA |≡ K_s', RB | ≡ K_s', $SAC|\equiv B|\equiv C(B)$

Here, $\boldsymbol{SAC|\equiv B|\equiv C(B)}$can be deduced by the same way as protocol 1. So the first goal is achieved.

Next, for second goal.

That is, we should prove that SAC believe B has accepted to the session, SAC | ≡ SP(B, S).

From the message ③, SA |≡ $\#N_2$ and the Rule 2, we can obtain SAC| ≡ #[ID_A, ID_B, SP(B, S), ID_s, N_2]. ……(i)

Then from SAC| ≡ A $\overset{K_s^+}{\leftrightarrow}$ SAC, the message③, (i) and Rule 6, we can deduce

SAC| ≡ A| ≡ [ID_A, ID_B, $SP(B,S)$, ID_s, N_2]. ……(ii)

from (ii), we can obtain SAC |≡ A| ≡ SP(B, S) by Rule2. ……(iii)

From the assumption SAC|≡ A| ⟹ SP(B, S)and Rule 3, we can deduce**SAC**| ≡ **SP**(**B**, **S**).

Finally, for the third goal. That is, we should prove that a new session key is to be generated and shared among all the session members.

From the message 4 and A| ≡ $\#N_2$, it follows that A| ≡ #[K_s', N_1+1] by Rule 2.. (iv)

Then from the message 5 and (iv),it yields that A| ≡ A $\overset{K_s^+}{\leftrightarrow}$ SAC by Rule 5. …….(v)

So A believes that Ks is secure.

From (iv), (v) and the message 5, we have A| ≡ SAC| ≡ K_s' by Rule 6. ……..(vi)

Consequently, from the assumption A|≡ SAC| ⟹ K_s' , (vi) and Rule 3, it follows that **A| ≡ $\mathbf{K_s'}$**.

That is, the new key Ks' has been shared with A.

$\mathbf{RA}| \equiv \mathbf{K}_{\mathbf{s}}'$and $\mathbf{RB}| \equiv \mathbf{K}_{\mathbf{s}}'$ can also be deduced similarly.

In turn, from SAC|≡↑ Pub(SAC), BA| ≡↑ Pub(B), we can obtain $B| \equiv B \overset{K_{(B,\ SAC)}^{-}}{\longleftrightarrow} SAC$ by Rule 4. ….(vii)

Then from $B| \equiv B \overset{K_{(B,\ SAC)}^{-}}{\longleftrightarrow} SAC$, $B| \equiv \#N_1$ and the message 5, it yields that $B| \equiv B \overset{K_{(B,\ SAC)}^{+}}{\longleftrightarrow} SAC$. ….(viii)

Consequently, from (viii) and $B| \equiv \#N_1$ and the message 5, we can deduce $B| \equiv SAC| \equiv K_S'$. ……(ix)

Then from (ix), $B| \equiv SAC| \Longrightarrow K_S'$ and Rule 4, it yields that $\boldsymbol{B}| \equiv \boldsymbol{K_S'}$.

Hence the theorem.

So we have proved the following.

- SAC can verify the identity of user B.
- User A (the session manager) can add new user B to his/her own session S.
- When adding a new user, the session key is regenerated and distributed among all session users.

5.6.3 Protocol 3: Accepting a New Session User

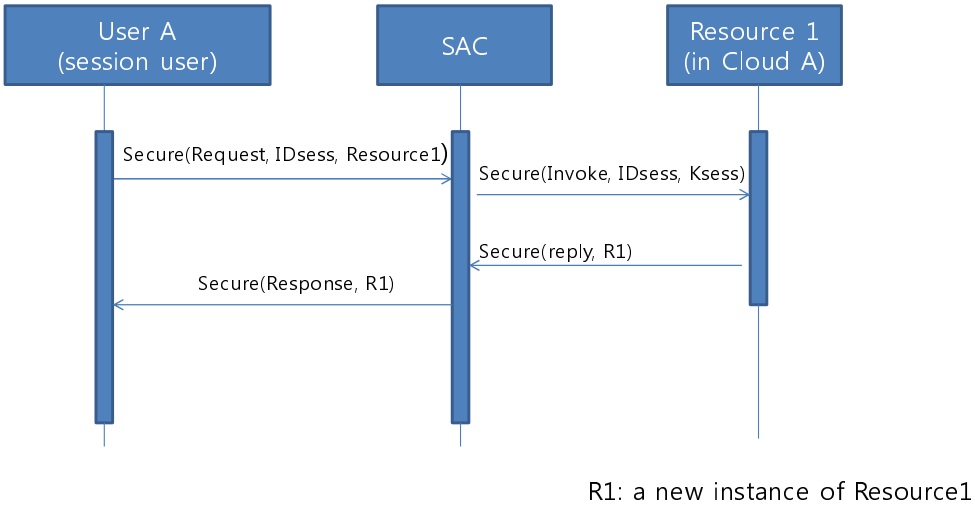

Figure 44: Protocol for accepting a new session user

Figure 44 illustrates the protocol for accepting a new session user. User A wants to call a new desired service, BI Resource1, in Cloud A (available in the session). First, User A sends an accepting request with IDsess and Resource1 to SAC. Then SAC invokes Resource1 with IDsess and Ksess. Resource1 spawns a new instance, R1, and delivers IDsess and Ksess to it. After this, Resource1 sends a reply with the identifier R1. SAC then registers it in the session and sends a response with R1 to User A. Then User A can call R1. The messages between SAC and User A are secured by session key Ksess, which has been previously shared among them. In addition, the messages between SAC and Resource1 are secured by the key shared using the Diffie-Hellman algorithm.

The following are shown messages transported in this protocol.

① $\mathbf{A} \rightarrow \mathbf{SAC} \quad [ID_A,\ ID_{SAC}, \mathbf{SP(R1, S)},\ N_1]_{K_s}$

② $\mathbf{SAC} \rightarrow \mathbf{R1} \quad [ID_{SAC},\ ID_{R1},\ ID_S,\ \mathbf{Ks},\ N_2]_{K_{SAC,R1}}$

③ $\mathbf{R1} \rightarrow \mathbf{SAC} \quad [ID_{R1},\ ID_{SAC},\ ID_{r1},\ N_2 + 1]_{K_{SAC,R1}}$

④ $\mathbf{SAC} \rightarrow \mathbf{A} \quad [ID_{SAC},\ ID_A,\ ID_{r1},\ N_1 + 1]_{K_S}$

When the user A in the session tried to contact Resource1 (R1), A first sends a request to SAC (message (1)). Then SAC sends a request to R1 (message (2)). R1 then generates a new instance (r1) of service and sends back the information about R1 to SAC. After receiving the response from R1 (message (3)), the SAC adds R1 to S and sends back a response to A (message (4)). The integrity of the messages (1) and (4) are protected by the secret key generated with DH algorithm and the integrity of the messages (2) and (3) are protected by the secret key shared within S.

The security goals in this protocol include the followings.

1) Accepting R1 as a member of session S.
2) Sharing the session key of session S, Ks, with R1

The goals are formally described as follows:

$\mathbf{SAC|} \equiv \mathbf{SP(R1, S)}$: SAC believed that R1 is accepted as a member of session S

$\mathbf{R1|} \equiv \mathbf{Ks}$ This means that session key must be shared with R1.

Assumptions:

$\mathbf{A|} \equiv \mathbf{A} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC:}$ A believes that SAC and A have the same session key and can use it to exchange of messages with SAC.

$\mathbf{SAC|} \equiv \mathbf{A} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC:}$ SAC believes that SAC and A has the same session key and can use it to exchange of messages with A.

$\mathbf{R1|} \equiv\uparrow \mathbf{Pub(R1)}$, $\boldsymbol{SAC|}\equiv\uparrow \mathbf{Pub(SAC)}$, $\boldsymbol{R1|} \equiv\uparrow \mathbf{Pub(SAC)}$, $\mathbf{SAC|} \equiv\uparrow \mathbf{Pub(R1)}$:

Users believe their private keys remain secure.

$\mathbf{SAC|} \equiv \mathbf{A|} \Rightarrow \mathbf{SP(R1, S)}$: SAC believes A can control accepting R1 as session member of the session S.

$\boldsymbol{R1|} \equiv \mathbf{SAC} \mid \Longrightarrow \boldsymbol{K_S}$: R1 believe SAC can control session key **Ks**.

$\boldsymbol{SAC|} \equiv \#\boldsymbol{N_1}$, $\boldsymbol{SAC|} \equiv \#\boldsymbol{N_2}$, SAC believes the nonces N_1, N_2 are fresh.

$\boldsymbol{A|}\equiv \#\boldsymbol{N_1}$, $\boldsymbol{R1|} \equiv \#\boldsymbol{N_2}$,

Proof. It is needed to deduce $\boldsymbol{SAC|} \equiv \boldsymbol{SP(R1, S)}$ and $\boldsymbol{R1|} \equiv \boldsymbol{Ks}$ from the assumputions of the protocol.

We obtain **SAC sees** $[\boldsymbol{ID_A}, \boldsymbol{ID_{SAC}}, \mathbf{SP(R1, S)}, \boldsymbol{N_1}]$ by the assumption $\mathbf{SAC|} \equiv \mathbf{A} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC,}$ the **message1** and **Rule 7**. …(1)

From the assumption $\boldsymbol{SAC|} \equiv \#\boldsymbol{N_1}$ and **Rule2**, it follows that $\boldsymbol{SAC|} \equiv \#[\boldsymbol{ID_A},\ \boldsymbol{ID_{SAC}}, \mathbf{SP(R1,S)},\ \boldsymbol{N_1}]$.

...(2)

And from (1), (2), the message1 and Rule 6, it yields $\boldsymbol{SAC|} \equiv \mathbf{A|} \equiv [\boldsymbol{ID_A},\ \boldsymbol{ID_{SAC}}, \mathbf{SP(R1,S)},\ \boldsymbol{N_1}]$.

…(3)

From (3) and Rule 1, it follows that $\boldsymbol{SAC|} \equiv \mathbf{A|} \equiv \mathbf{SP(R1,S)}$. …(4)

So we can deduce that $\mathbf{SAC|} \equiv \mathbf{SP(R1,S)}$ by (4), the assumption $\mathbf{SAC|} \equiv \mathbf{A|} \Rightarrow \mathbf{SP(R1,S)}$ and Rule 3.

Then from the assumption $\mathbf{R1|} \equiv\uparrow \mathbf{Pub(SAC)}$, $\mathbf{R1|} \equiv\uparrow \mathbf{Pub(R1)}$, we can obtain

$\mathbf{R1|} \equiv \mathbf{R1} \overset{K_{R1,SAC}}{\longleftrightarrow} \mathbf{SAC}$ by **Rule 4**. ...(5)

From (5), the message2 and Rule 7, it yields R1 sees $[\boldsymbol{ID_{SAC}},\ \boldsymbol{ID_{R1}},\ \boldsymbol{ID_S},\ \mathbf{Ks},\ \boldsymbol{N_2}]$.

…(6)

From the assumption $\boldsymbol{R1|} \equiv \#\boldsymbol{N_2}$, it follows that $\boldsymbol{R1|} \equiv \#[\boldsymbol{ID_{SAC}},\ \boldsymbol{ID_{R1}},\ \boldsymbol{ID_S},\ \mathbf{Ks},\ \boldsymbol{N_2}]$ by Rule 2. …(7)

From (5), (6), (7) and Rule 6, it yields that $\boldsymbol{R1|} \equiv \mathbf{SAC}\ | \equiv [\boldsymbol{ID_{SAC}},\ \boldsymbol{ID_{R1}},\ \boldsymbol{ID_S},\ \mathbf{Ks},\ \boldsymbol{N_2}]$. …(8)

And we obtain $\boldsymbol{R1|} \equiv \mathbf{SAC}\ | \equiv \mathbf{Ks}$ by (8) and Rule 1. …(9)

Therefore we can deduce that $\boldsymbol{R1|} \equiv \mathbf{Ks}$ by (9), the assumption $\boldsymbol{R1|} \equiv \mathbf{SAC}\ | \Longrightarrow \boldsymbol{K_S}$ and Rule 3.

Hence the theorem.

So we have proved the following:

- SAC can believe that user A wants to a new member R1 to the session.
- When adding a new member, the session key is to be shared with the new member securely.

5.6.4 Protocol 4: Leaving a Session

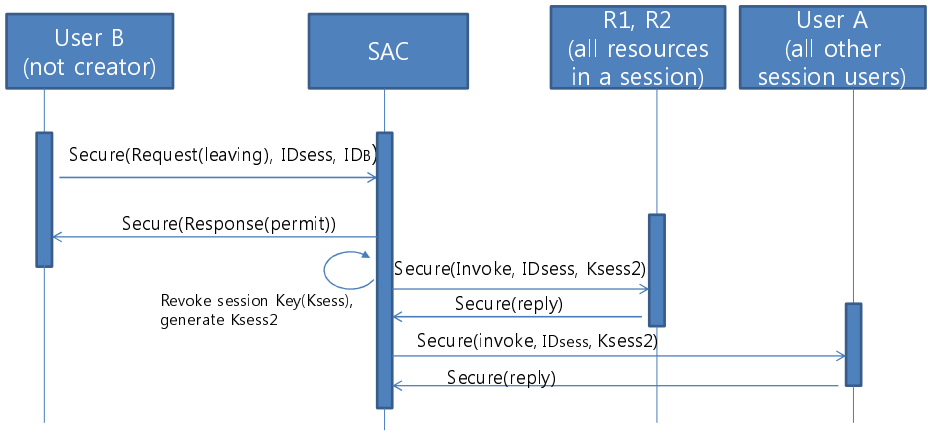

Figure 45: Protocol for leaving a session

Figure 45 illustrates the protocol for leaving a session. User B, who is a session user but not the creator, wants to leave a session. First, User B sends a request to SAC to leave the session. After User B receives a response of "permit" from SAC, User B leaves the session. Next, after SAC removes User B from the session user list, revokes the old session key and invokes a new session key. Then SAC broadcasts it to all the other session users and to all the resources that belong to the session.

The messages in this protocol are shown below.

① B→ SAC $[ID_B,\ ID_{SAC},\ \mathbf{Exit(B,S)},\ N_1]_{K_s}$

② SAC→ B $[ID_{SAC},\ ID_B,\ N_1+1]_{K_S}$

③ SAC→ R1 $[ID_{SAC}, ID_{R1},\ ID_S,\ \mathbf{Ks'},\ N_2]_{K_{SAC,R1}}$

④ R1→ SAC $[\ ID_{R1}, ID_{SAC},\ N_2+1]_{K_{SAC,R1}}$

⑤ SAC→ A $[ID_{SAC},\ ID_A,\ \ ID_S,\ \mathbf{Ks'},\ N_3]_{K_{SAC,A}}$

⑥ A→ SAC $[\ ID_A, ID_{SAC},\ N_3+1]_{K_{SAC,A}}$

When B (not session creator) tries to leave a session, B first sends a request to SAC (message(1)). Then SAC removes B from the session S and sends back a response of permit to B (message (2)). And then SAC regenerates the session key and informs all session members except B of updating the session key (message (3), (5)). The members send back responses to SAC (message (4), (6)). The integrity of message (1) and (2) are protected by the shared secret key within S and the integrity of messages (3), (4), (5) and (6) are protected by the secret keys generated with D-H algorithm.

The security goals in this protocol include the following:

1) Removing B from the session S
2) Regenerating session key and sharing it among the members of session S except B.

The goals are formally described as follows.

$\mathbf{SAC|} \equiv \mathbf{Exit(B,S)}$: SAC believed that B left the session S.

$\mathbf{R1|} \equiv \mathbf{Ks'}$ $\mathbf{A|} \equiv \mathbf{Ks'}$ this means that a new session key Ks' must be shared among the session members.

Assumptions:

$\mathbf{SAC|} \equiv \mathbf{B} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC:}$ SAC believe that SAC and B have the same session key and can use it to exchange of messages with B.

$\mathbf{R1|} \equiv\uparrow \mathbf{Pub(R1)}$, $\boldsymbol{R1|}\equiv\uparrow \mathbf{Pub(SAC)}$, $\boldsymbol{A|} \equiv\uparrow \mathbf{Pub(A)}$, $\mathbf{A|} \equiv\uparrow \mathbf{Pub(SAC)}$: Users believe their private keys remain secure.

$\mathbf{SAC|} \equiv \mathbf{B|} \Rightarrow \mathbf{Exit(B,S)}$: SAC believes B can control the leaving the session S.

$\boldsymbol{R1|} \equiv \mathbf{SAC} \mid \Longrightarrow \boldsymbol{K_S'}$: R1 believes SAC can control session key **Ks'**.

$\boldsymbol{SAC|} \equiv \#\boldsymbol{N_1}$, $\boldsymbol{SAC|} \equiv \#\boldsymbol{N_2}$, SAC believes the nonces N_1, N_2 are fresh.

$\boldsymbol{A|}\equiv \#\boldsymbol{N_3}$, $\boldsymbol{R1|} \equiv \#\boldsymbol{N_2}$,

Proof. It is needed to deduce $\boldsymbol{SAC| \equiv Exit(R1, S)}$, $\boldsymbol{A| \equiv Ks'}$ ***and*** $\boldsymbol{R1| \equiv Ks'}$***from the assumputions of the protocol.***

We obtain SAC sees $[\boldsymbol{ID_B},\ \boldsymbol{ID_{SAC}},\ \mathbf{Exit(B,S)},\ \boldsymbol{N_1}]$ by the assumption $\mathbf{SAC| \equiv B} \overset{\boldsymbol{K_S^+}}{\leftrightarrow} \mathbf{SAC}$, the message1 and Rule 7. …(1)

From the assumption $\boldsymbol{SAC| \equiv \#N_1}$ and Rule2, it follows that $\boldsymbol{SAC| \equiv \#[ID_B,\ ID_{SAC},}$ $\mathbf{Exit(B,S)},\ \boldsymbol{N_1}]$. …(2)

From (1), (2), the message1 and Rule 6, it yields that $\boldsymbol{SAC| \equiv B| \equiv [ID_B,\ ID_{SAC},}$ $\mathbf{Exit(B,S)},\ \boldsymbol{N_1}]$ …(3)

From (3) and Rule 1, it follows that $\boldsymbol{SAC| \equiv B| \equiv}\ \mathbf{Exit(B,S)}$. …(4)

So we can deduce that $\mathbf{SAC| \equiv Exit(B,S)}$ by (4), the assumption $\mathbf{SAC| \equiv B| \Rightarrow}$ $\mathbf{Exit(B,S)}$ and Rule 3.

Next, from the assumption $\mathbf{R1| \equiv\uparrow Pub(SAC)}$, $\mathbf{R1| \equiv\uparrow Pub(R1)}$, we can obtain

$\mathbf{R1| \equiv R1} \overset{\boldsymbol{K_{R1,SAC}^-}}{\longleftrightarrow} \mathbf{SAC}$ by **Rule 4**. …(5)

From (5), the message3 and Rule 7, it yields R1 sees $[\boldsymbol{ID_{SAC}, ID_{R1}},\ \boldsymbol{ID_S},\ \mathbf{Ks'},\ \boldsymbol{N_2}]$. …(6)

Then from the assumption $\boldsymbol{R1| \equiv \#N_2}$, it follows that $\boldsymbol{R1| \equiv \#[ID_{SAC}, ID_{R1}},\ \boldsymbol{ID_S},$ $\mathbf{Ks'},\ \boldsymbol{N_2}]$ by Rule2. …(7)

From (5), (6), (7) and Rule 6, it follows that $\boldsymbol{R1| \equiv SAC| \equiv [ID_{SAC}, ID_{R1}},\ \boldsymbol{ID_S},\ \mathbf{Ks'},$ $\boldsymbol{N_2}]$. …(8)

And then we obtain $\boldsymbol{R1| \equiv SAC| \equiv Ks'}$ by (8) and Rule 1. …(9)

Therefore, we can deduce that $\boldsymbol{R1| \equiv}$ **Ks'** by **(9)**, the assumption $\boldsymbol{R1| \equiv SAC| \Longrightarrow K_S'}$ and Rule 3.

And $\boldsymbol{A| \equiv}$ **Ks'** can be deduced by the same way as above.

Hence the theorem.

So we have proved the following:

- SAC can believe that user B wants to leave the session S.
- Whenever a session user leaves the session, the session key is regenerated and distributed among all other session users.

5.6.5 Protocol 5: Ending a session

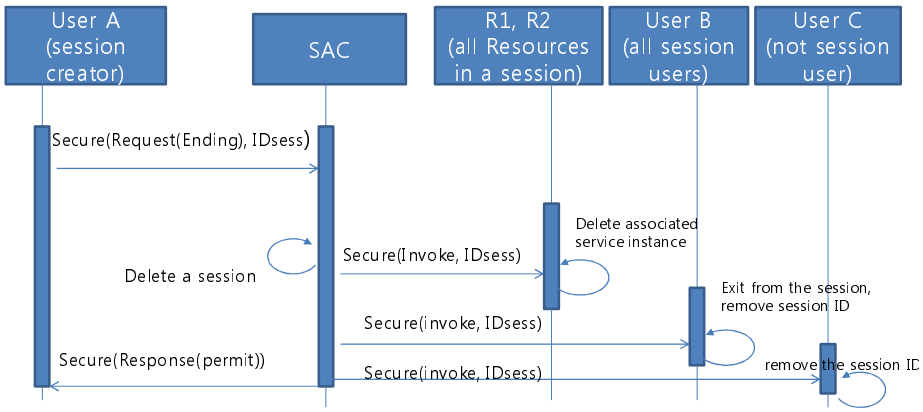

Figure 46: Protocol for ending a session

Figure 46 illustrates the protocol for ending a session. User A (session creator) wants to end his/her session. First User A sends a request to SAC to end the session. Then SAC removes the session from the session list and broadcasts, to all users (session users and other users) and resources that belong to the session, that the session has been removed. Next, SAC sends a response to User A, and then User A ends his/her session. After receiving "end session" messages, session users revoke the session key, exit from that session and remove the session ID from their session ID lists. The other users also remove the session ID. The resources terminate the corresponding service instances.

The messages in this protocol are shown as followings.

❶. **A→ SAC** $[ID_A, ID_{SAC}, \mathbf{Exit(A,S)}, N_1]_{K_s}$

❷. **SAC→ R1** $[ID_{SAC}, ID_{R1}, \mathbf{Rm(S)}, N_2]_{K_S}$

❸. **SAC→ B** $[ID_{SAC}, ID_B, \mathbf{Rm(S)}, N_3]_{K_S}$

④. SAC→ C $[ID_{SAC}, ID_C, \mathbf{Rm(S)}, N_4]_{K_{SAC,C}}$

⑤. SAC→ A $[ID_{SAC}, ID_A, N_1 + 1]_{K_S}$

When A (session creator) tries to end a session, A first sends a request to SAC (message (1)).

Then SAC removes all the resources in S from S and informs them of ending a session (message (2)).

In turn, SAC removes all the users in S from S and informs them of ending a session (message (3)). Then SAC informs other users, connecting to itself, of ending a session (message (4)).

Finally, the SAC expires the session key, removes the session S and sends back a response to A.

The integrity of messages (1), (2), (3) and (5) are protected by the shared secret key with S and the integrity of message (4) is protected by the secret key generated with D-H algorithm.

The security goals in this protocol include the following:

1) Removing session S from the session list
2) Informing all user of Removing the session

The goals are formally described as following:

$\mathbf{SAC|} \equiv \mathbf{Exit(B,S)}$: SAC believed that the session creator A left the session S.

$\mathbf{R1|} \equiv \mathbf{Rm(S)}$ $\mathbf{A|} \equiv \mathbf{Rm(S)}$ $\mathbf{C|} \equiv \mathbf{Rm(S)}$ this means that a new session key Ks' must be shared among the session members. Here A and R1 are the members of the session S, C is other user that connected to SAC.

Assumptions:

$\mathbf{SAC|} \equiv \mathbf{A} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC:}$ SAC believes that SAC and A has the same session key and can use it to exchange of messages with A.

$\mathbf{R1|} \equiv \mathbf{R1} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC:}$ R1 believes that SAC and R1 has the same session key and can use it to exchange of messages with SAC.

$\mathbf{B|} \equiv \mathbf{B} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC:}$ B believes that SAC and B has the same session key and can use it to exchange of messages with SAC.

$\mathbf{C|} \equiv\uparrow \mathbf{Pub(C)}$, $\mathbf{C|} \equiv\uparrow \mathbf{Pub(SAC):}$ User C believes the private keys remain secure.

$\mathbf{SAC|} \equiv \mathbf{A|} \Rightarrow \mathbf{Exit(A,S)}$: SAC believes A can control the leaving the session S (leaving of creator means ending a session)

$\boldsymbol{R1|} \equiv \mathbf{SAC} \mid \Longrightarrow \mathbf{Rm(S)}$: R1 believes SAC can control removing the session S.

$\boldsymbol{B|} \equiv \mathbf{SAC} \mid \Longrightarrow \mathbf{Rm(S)}$, $\boldsymbol{C|} \equiv \mathbf{SAC} \mid \Longrightarrow \mathbf{Rm(S)}$

$\boldsymbol{SAC|} \equiv \#\boldsymbol{N_1}$ SAC believes the nonces $\mathbf{N_1}$, $\mathbf{N_2}$ are fresh.

$\boldsymbol{R1|} \equiv \#\boldsymbol{N_2}$, $\boldsymbol{B|} \equiv \#\boldsymbol{N_3}$, $\boldsymbol{C|} \equiv \#\boldsymbol{N_4}$

Proof. It is needed to deduce $\boldsymbol{SAC|} \equiv \boldsymbol{Exit(A,S)}$, $\boldsymbol{R1|} \equiv \boldsymbol{Rm(S)}$, $\boldsymbol{B|} \equiv \boldsymbol{Rm(S)}$ ***and*** $\boldsymbol{C|} \equiv \boldsymbol{Rm(S)}$ ***from the assumptions of the protocol.***

Firstly, we obtain SAC sees [$\boldsymbol{ID_A}$, $\boldsymbol{ID_{SAC}}$, $\mathbf{Exit(A,S)}$, $\boldsymbol{N_1}$] by the assumption $\mathbf{SAC|} \equiv \mathbf{A} \overset{K_S^+}{\leftrightarrow} \mathbf{SAC}$, the message⊕ and Rule 7. …(1)

From the assumption $\boldsymbol{SAC|} \equiv \#\boldsymbol{N_1}$ and Rule2, it follows that $\boldsymbol{SAC|} \equiv \#[\boldsymbol{ID_A}, \boldsymbol{ID_{SAC}}, \mathbf{Exit(A,S)}, \boldsymbol{N_1}]$. …(2)

From (1), (2), the message1 and Rule 6, it follows $\boldsymbol{SAC|} \equiv \mathbf{A|} \equiv [\boldsymbol{ID_A}, \boldsymbol{ID_{SAC}}, \mathbf{Exit(A,S)}, \boldsymbol{N_1}]$ …(3)

Then from (3) and Rule 1, it follows that $\boldsymbol{SAC|} \equiv \mathbf{A|} \equiv \mathbf{Exit(A,S)}$. …(4)

Consequently, we can deduce that $\mathbf{SAC|} \equiv \mathbf{Exit(A,S)}$ by (4), the assumption $\mathbf{SAC|} \equiv \mathbf{B|} \Rightarrow \mathbf{Exit(B,S)}$ and Rule 3.

Next, we obtain R1 sees $[\boldsymbol{ID_{SAC}}, \boldsymbol{ID_{R1}}, \mathbf{Rm(S)}, \boldsymbol{N_2}]$ by the assumption $\mathbf{R1|} \equiv \mathbf{R1} \overset{\boldsymbol{K_S^+}}{\leftrightarrow} \mathbf{SAC}$, the message2 and Rule 7. …(5)

From the assumption $\boldsymbol{R1|} \equiv \#\boldsymbol{N_2}$ and Rule2, it yields that $\boldsymbol{R1|} \equiv \#[\boldsymbol{ID_{SAC}}, \boldsymbol{ID_{R1}}, \mathbf{Rm(S)}, \boldsymbol{N_2}]$. …(6)

Then from (5), (6), the message2 and Rule 6, it follows $\boldsymbol{R1|} \equiv \mathbf{SAC|} \equiv [\boldsymbol{ID_{SAC}}, \boldsymbol{ID_{R1}}, \mathbf{Rm(S)}, \boldsymbol{N_2}]$ …(7)

From (7) and Rule 1, it follows that $\boldsymbol{R1|} \equiv \mathbf{SAC|} \equiv \mathbf{Rm(S)}$. …(8)

Consequently, we can deduce that $\mathbf{R1|} \equiv \mathbf{Rm(S)}$ by (8), the assumption $\mathbf{R1|} \equiv \mathbf{SAC|} \Rightarrow \mathbf{Rm(S)}$ and Rule 3.

$\mathbf{B|} \equiv \mathbf{Rm(S)}$ can be deduced as the same way as above.

And then, from the assumption $\mathbf{C|} \equiv\uparrow \mathbf{Pub(SAC)}$, $\mathbf{C|} \equiv\uparrow \mathbf{Pub(C)}$, we can obtain

$\mathbf{C|} \equiv \mathbf{C} \overset{\boldsymbol{K_{C,SAC}^-}}{\longleftrightarrow} \mathbf{SAC}$ by Rule 4. …(9)

From (9) and the message4, it follows $\mathbf{C\ sees\ [}\boldsymbol{ID_{SAC}}, \boldsymbol{ID_C}, \mathbf{Rm(S)}, \boldsymbol{N_4}\mathbf{]}$ by Rule 7. …(10)

Then from the assumption $\boldsymbol{C|} \equiv \#\boldsymbol{N_4}$, it follows that $\boldsymbol{C|} \equiv \#[\boldsymbol{ID_{SAC}}, \boldsymbol{ID_C}, \mathbf{Rm(S)}, \boldsymbol{N_4}]$ by Rule 2. …(11)

Consequently (9), (10), (11) and Rule6, it follows that $\boldsymbol{C|} \equiv \mathbf{SAC|} \equiv [\boldsymbol{ID_{SAC}}, \boldsymbol{ID_C}, \mathbf{Rm(S)}, \boldsymbol{N_4}]$. …(12)

And then we obtain $\boldsymbol{C|} \equiv \mathbf{SAC|} \equiv \mathbf{Rm(S)}$ by (12) and Rule 1. …(13)

Therefore, we can deduce that $\boldsymbol{C|} \equiv \mathbf{Rm(S)}$ by (13), the assumption $\boldsymbol{C|} \equiv \mathbf{SAC|} \Rightarrow \mathbf{Rm(S)}$ and Rule 3.

Hence the theorem.

So we have proved the following.

- SAC can believe that user A who is a session creator wants to end the session S.

- When a session is ended, all users can be notified of the fact.

5.7 **Analytic Assessment**

In this section, we focus on the analytic assessment of the MPA protocols and the key management mechanisms for business sessions.

5.7.1 Analysis of the Protocols

The Multiparty Authentication System can help session users authenticate their session memberships to simplify the authentication processes within multiparty sessions.

If a user intends to create a new session, it has to invoke the Session Authority Cloud (SAC), when an authentication process between SAC and the user needs to be performed and authentication processes between SAC and the services that the user (session creator) wants to include in its session need to be performed.

Also if a new user intends to join a session, it has to invoke the SAC and may join the session under the approval of the session creator. At this time, an authentication process between the user and SAC is to be performed.

In order to facilitate the analysis benefits that the proposed system may introduce to multiparty sessions, assume that there is a multiparty session on cloud S which consists of n session users $U_1, U_2, \ldots U_n$ and m services $V_1, V_2, \ldots, V_m$. The multiparty session S is under the management of the SAC.

The number of services that U_i intends to invoke to is denoted as c_i.

Without this system, there are $N_0 = \sum_{i=0}^{n} c_i$ authentication processes to be performed in S.

And there are $N_1 = n + m$ authentication processes between SAC and session members (users and services).

Therefore, there are totally $N_{total} = N_0 + N_1$ authentication processes performed in S. Among these authentication processes, there are N_0 authentication processes (between users and services) and N_1 authentication processes (between SAC and session users).

The number of authentication processes between users and services simplified by our system is denoted as N_{opt}.

5.7.1.1 Worst case scenario

With our system, the authentication processes between SAC and session members (users and services). Figure 47, illustrates a worst case scenario (1) where there is only one user and m services in session (S).

Here, $N_0 = m$, $N_1 = 1 + m$, $N_{total} = 2m + 1$ and $N_{opt} = 0$.

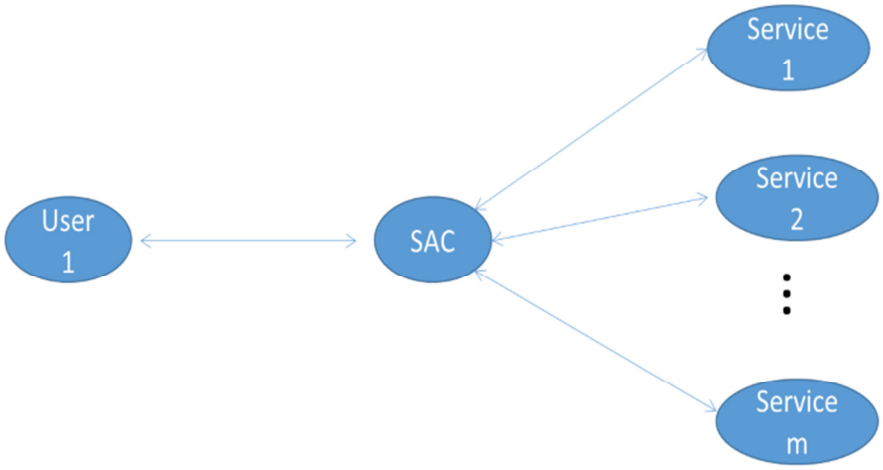

Figure 47: A worst case scenario-1 (session approval)

Figure 48, illustrates a worst case scenario (2). Other worst case scenarios exist when there are either n users and one service or where there are 2 users and 2 services. In these cases, the authentication process cannot be simplified and the SAC process offers no significant advantage over direct authentication. In addition, the two-party session technique

does not address the issue of different Cross-Realm Authentication, which requires credential conversion and the establishment of authentication paths.

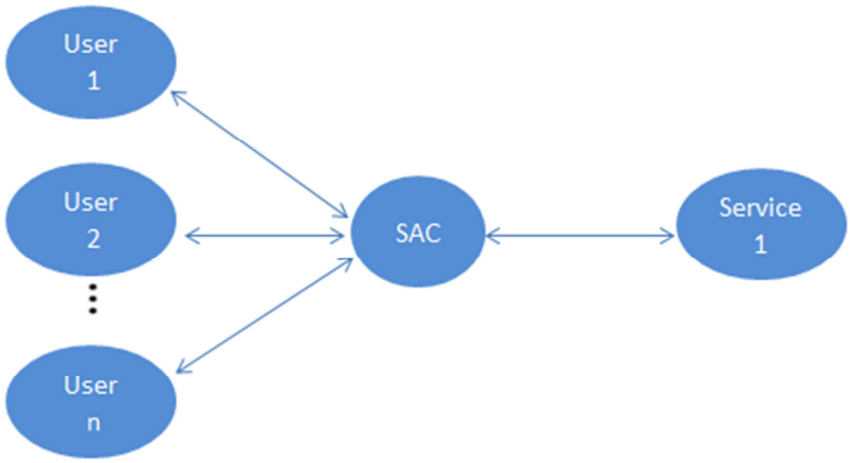

Figure 48: A worst case scenario-2

5.7.1.2 Best case scenario

Figure 49 illustrates a best case scenario. In this case where there are n session users and m services, where both n and m are much greater than 2. The benefit is obtained since each user will be able to access all of the m services.

Here, $c_1 = c_2 = \cdots = c_n = m$, $N_0 = nm$ and $N_1 = n + m$. Thus, $N_{total} = nm + (n + m)$.

With our system, whenever a new user joins a session, an authentication process only is performed.

So, $N_0 = m$, $N_1 = n + m$, and $N_{total} = 2m + n$.

Thus, $N_{opt} = (n - 1)m$.

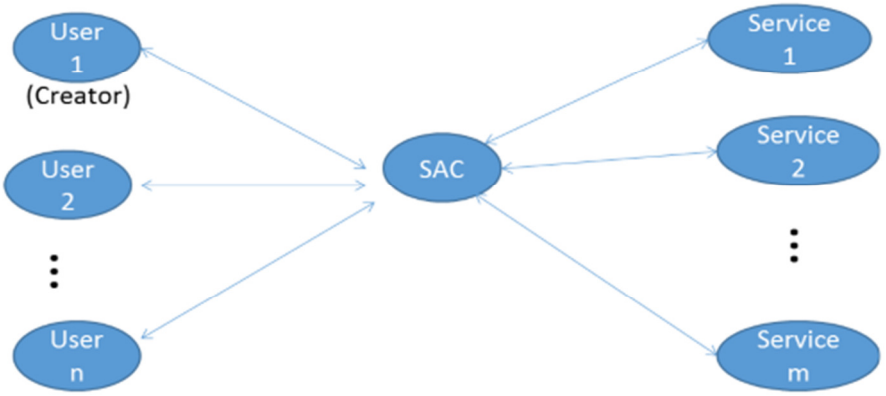

Figure 49: The best case scenario

Without our system, the authentication processes to be performed in session because each user has to be authorised by all the services that he/she wants to access. Therefore, the multiparty authentication system on cloud can introduce more benefits to a session when the number of users and services is increased.

5.7.2 Analysis of the Key Management

How to negotiate and distribute secret keys is always a critical concern during the design of authentication protocols. In practice, authentication can be performed in different ways, and different key management mechanisms can be employed. A secret key can be used to encrypt the messages transported within the session so that prevent the messages from being known or modified by the others from the outside of the session. With our system, whenever a new user join a session, a new session key is generated and shared with session members and the old key is expired before the user is added to a session. So the new user cannot know about the old key and the messages previously encrypted by it.

Whenever a session user leaves a session, a new session key is generated and shared with session members and the old key is expired as soon as the user has left the session. So the user cannot know the new session key and messages subsequently encrypted.

In our system, each user has an authentication channel linked with SAC, each channel has its one secret key. This channel can be used to verify whether the other side is a session member.

To facilitate the discussion, assume there are n session users $U_1, U_2, \ldots, U_n$ and m services, $V_1, V_2, \ldots, V_m$.

Without our system, each user needs to be maintain m secret key shared with services and one secret key shared with SAC. So there are (m+1)n secret keys in a session. These secret keys are managed by SAC.

The greater the number of session members is, the greater the number of keys exist and the more complex the management of keys becomes.

With our system, there are (m+n+1) keys in a session.

The figures below illustrate relationship of the numbers of users, services and secret keys.

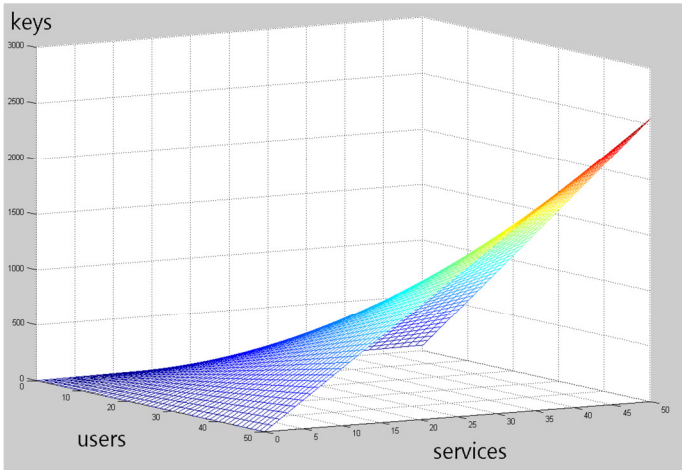

Figure 50: The case without employing multiparty authentication system on cloud

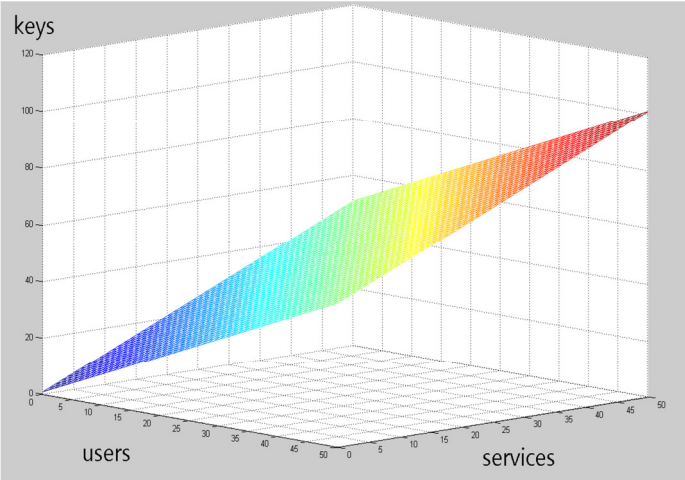

Figure 51: The case with employing multiparty authentication system on cloud

We can realise that the rate of increasing the number of keys is very fast without employing our system. Before an authentication protocol is applied in practice, it needs to be carefully examined in both correctness and performance. So this section introduces the work of analysing the multiparty authentication system on cloud.

Using BAN logic we have proved the perfection of our protocol. We can confirm follows with the method. The assumptions used to design Multi-party authentication system on cloud are re-examined. The objectives of our protocol are achievable. The crypto methods used in our protocol can prevent impersonation attacks and with the analytic assessment results, our protocol simplifies the authentication processes. In direct proportion to the increasing number of session memberships, if our solution is not used, the number of authentication processes increases and the authentication procedure becomes more complex. However, with our solution it can be simplified. Also with our solution the management of session key can be more readily facilitated.

5.8 **Evaluating and Presenting Analysis Results**

5.8.1 Analysis of results

The process of simulation execution is shown in the figure below. It may be observed that the simulation executed 178 million events in the operation of mere seven minutes and nine seconds on the network. This is because it is a reasonably large network with 1000 trusted principals accessing it. This is a near real-world scenario and hence the results have practical significance to a large extent.

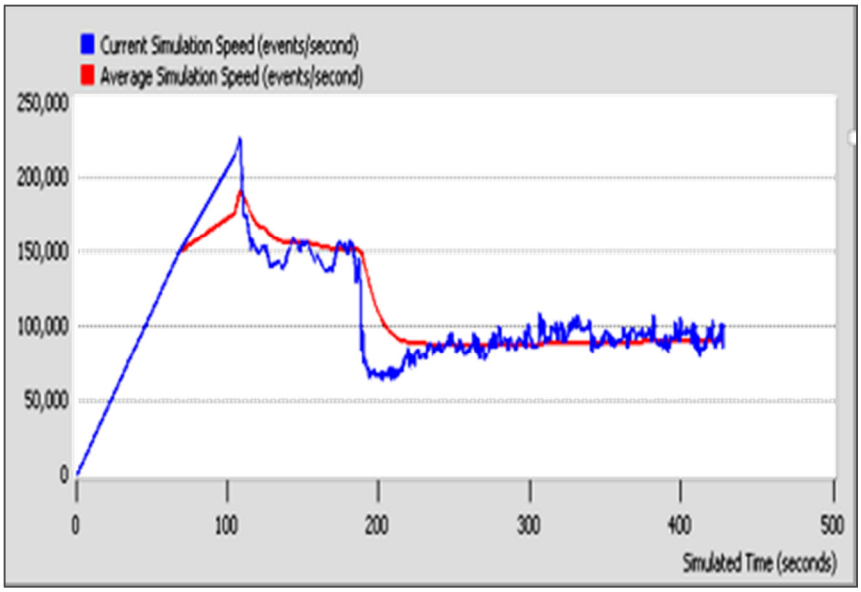

Figure 52: Executing the simulation

Figure 53 illustrates the TCP sessions initiated by the node "A". The active TCP sessions count exceeded 2000 during the simulation period indicating that each trusted principal has made two session requests on an average. Hence, the authentication protocol has been triggered more than 2000 times on the network.

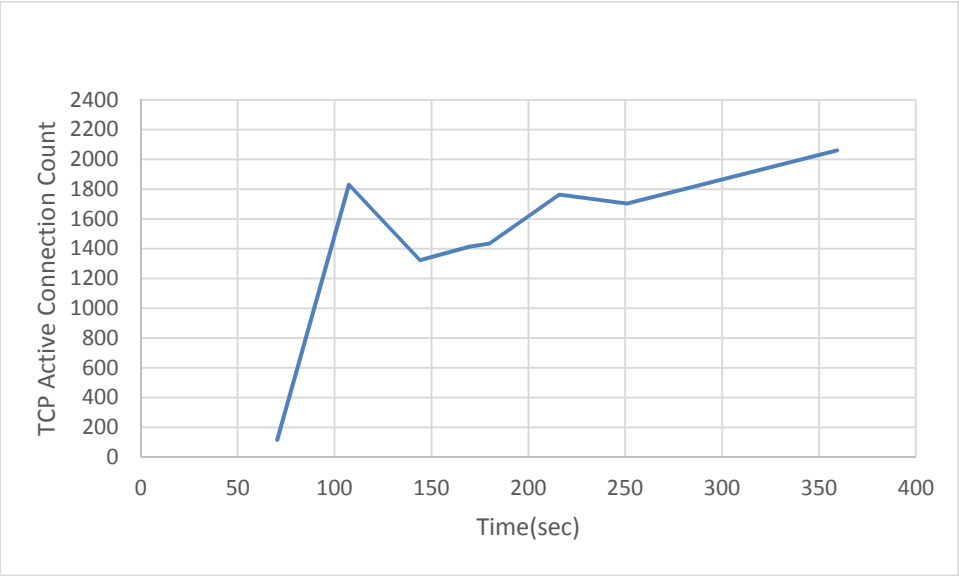

Figure 53: TCP sessions initiated by the node "A"

The overall performance and behaviour of authentication protocol tasks on the network are shown in Figure 54. The first statistic shows that the overall (end-to-end) response time of the authentication protocol on the network is about 60 seconds. This is genuine given the time taken in establishing the TCP connection and transferring the data. Hence, 60 seconds is a committed performance for executing all the 13 phases. However it is not infeasible knowing that it is only a one-time activity for each user being added by the trusted principal.

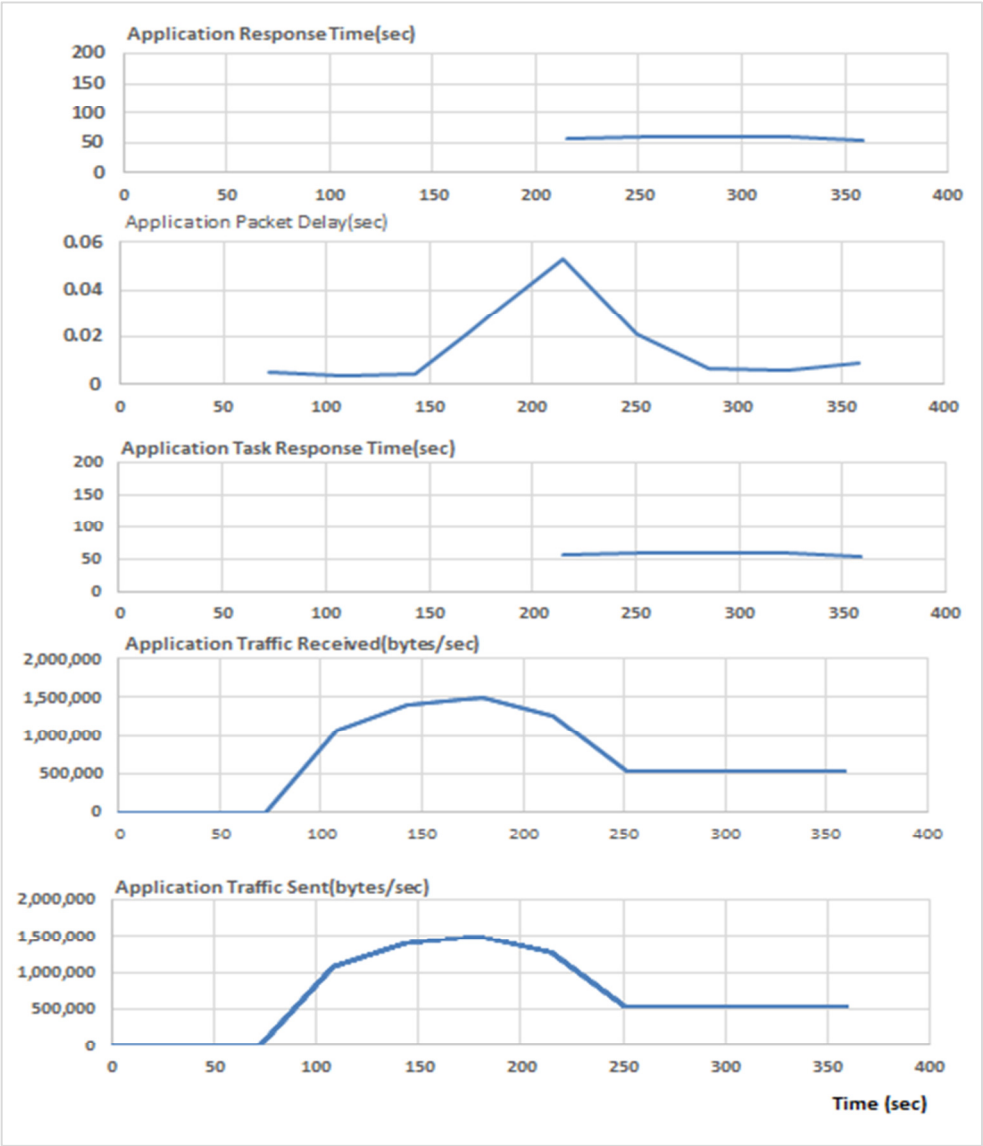

Figure 54: Overall performance metrics and behaviours of the authentication protocol tasks on the network

It is important to note that the delays because of network or host-based congestions are more serious than protocol execution delays. This has been verified by looking into the second statistic packet network delay. It is observed that the maximum delay occurred on the network is slightly less than 0.06 seconds. It was further confirmed by getting into device specific reports that none of the servers, switches, and links had registered any packet queues or packet forwarding delays on the network. Hence, 60 seconds is a committed performance

for executing all the 13 phases. This delay has occurred because of the amount of data exchanged per phase. The statistic "application task response time" is the same as "application phase response time" in this model because each phase has only one task in the algorithm. The last two statistics reflect the overall authentication traffic sent and received on the network.

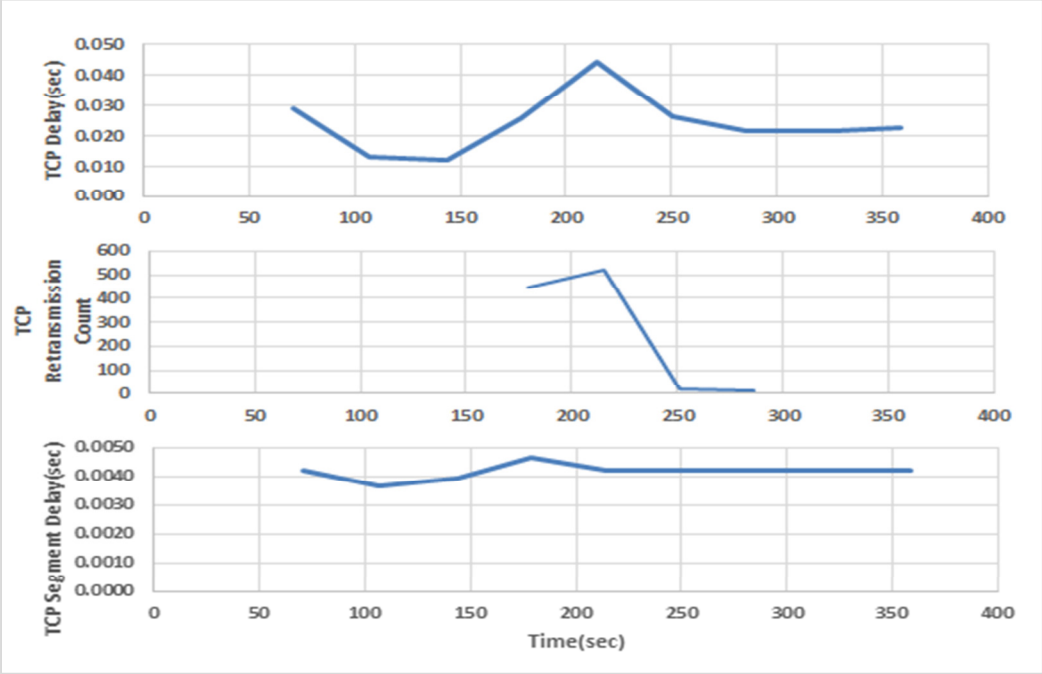

Figure 55: Delays investigated at the transport layer

For 1000 trusted principals interacting simultaneously, the traffic of up to 1.5 Mbps is quite moderate. This shows that the protocol is loading the network moderately. The delays have been further investigated at the transport layer, as well, as shown in Figure 55. There is hardly any TCP delay on the network. However, there have been up to 500 TCP retransmissions at one point when the traffic is at its peak. Five hundred TCP retransmissions out of 2000 TCP sessions is a considerable number necessitating network tuning.

Before making the final recommendations, the response times of individual phases of the authentication protocol are also investigated. Figure 54 illustrates this statistic for six out

of the eleven phases of the protocol. The overall response time is about 5 seconds per phase. This is genuine given the time taken in establishing the TCP connection and transferring the data.

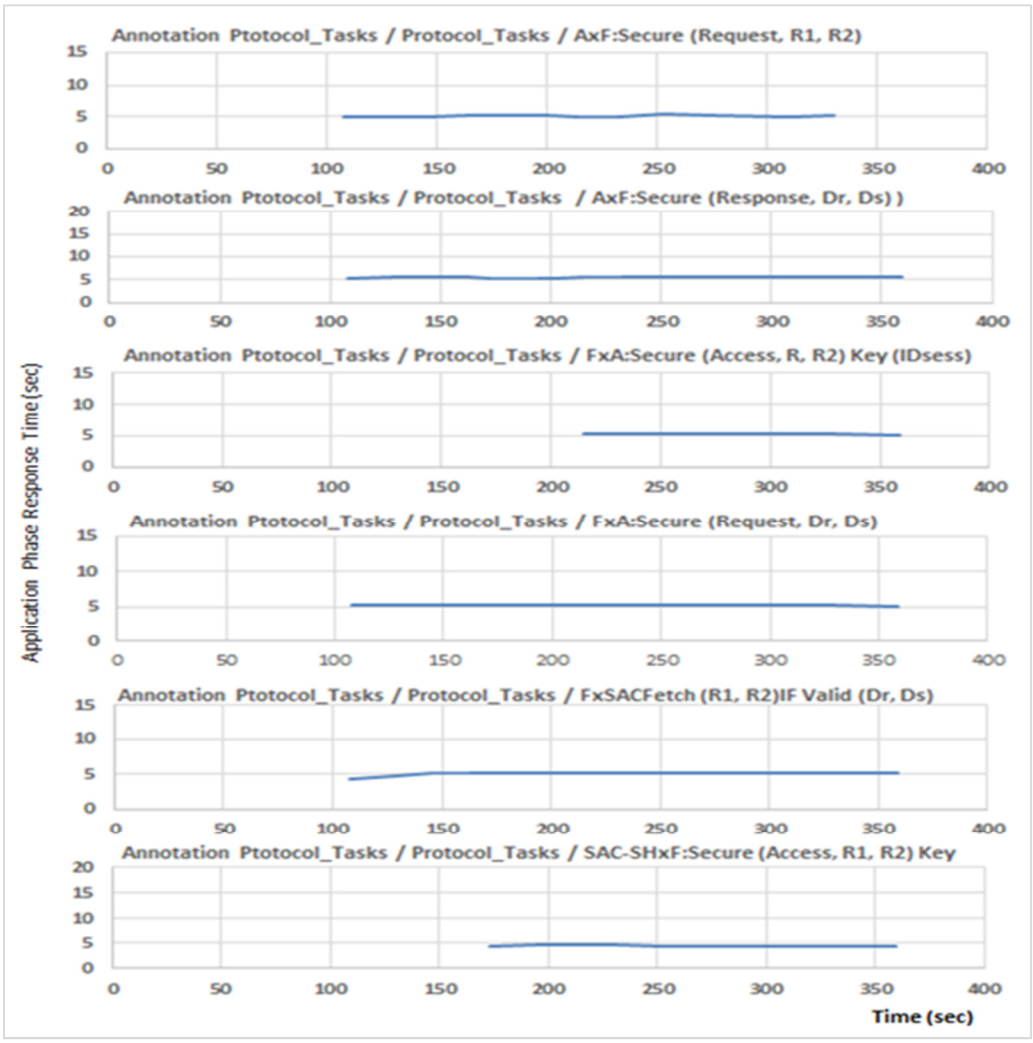

Figure 56: Response times of the individual phases of the authentication protocol

The above results are pertaining to the phases executed without a timeout. A second simulation was carried out by including a timeout of 60 seconds per phase. As per the previous results, the phases should not time out given that each phase was taking about 5 seconds to execute. However, as shown in Figure 57, the number of application instances reduced significantly as the phases of the protocol progressed. This indicates massive session

drops because of timeout configuration. This should not have happened because the timeout configured per phase was larger than the average phase duration observed. Multiple settings were tested but the results were similar. This is a problem needing further investigation. Perhaps, a version of OPNET modeller may be needed to investigate more deeply on what is happening when timeouts are configured. In this research, it is recommended that the 13-step authentication protocol should not have any timeout configured. However, a localised timeout can be configured at F.

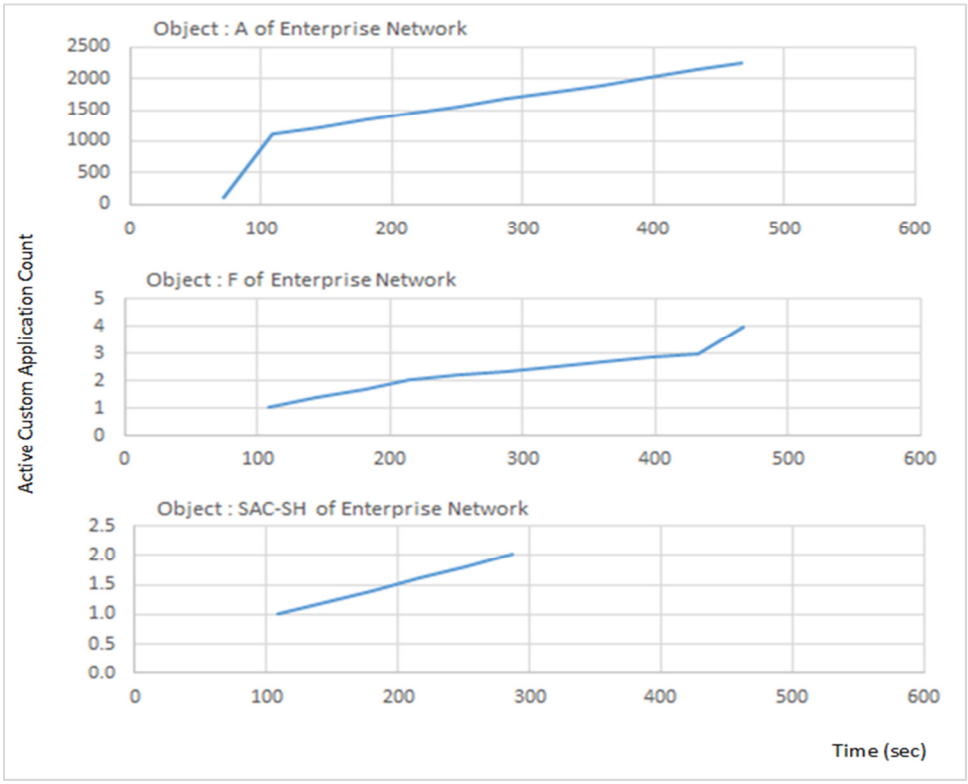

Figure 57: Indicating progressive reduction of instances count when introducing a timeout

In our approach, Cloud security service has a built-in relational database for examining the tenant sessions. The databases at the firewall (DB_{FW}) cloud help in authenticating, authorisation and providing access permission to the session initiated by the tenant. Tenant metadata (DB_{META}) comprises tenants' information for enabling authorisation

to designated application instances and database objects that the tenant has subscribed. Tenants' vault comprises keys and digital signatures stored in database objects (DB_{VAULT}) that can be retrieved based on tenant authorisation enabled by the metadata layer.

Distributed Denial of Service (DDoS) is one of the challenges that may arise in any network based systems. The DDoS attacks generally exist in two forms, a network-based attack that loads the service using a bandwidth, and an application-layer attack which overloads the service and a database with many application calls (Modi et al, 2013). The high volume of packets moving to the target creates a denial of service as the media focuses on the target of a DDoS attack. Denial of Service attacks can result in significant loss of service of any system. The attack master then identifies other vulnerable devices or processes in the network for potential attack. After infection, the attacker instructs the compromised devices to attack a single target on the network. In the network attack, the DDOS loads the network such that all available bandwidth is consumed and further communication is reduced to an impractically slow speed. The attack may take place as the attackers tries to break into the security control of the databases DBfw, DBmeta, and DBvault and gain access to the client data. A general sequence of events in an attack begins with the attacker targeting a cloud database. In such scenario, the attacker may gain access to a VM in the capacity of a verified tenant. The Cloud service providers offer a number of subscriptions and the tenant is given access to their virtualized environments after some preliminary verification (like ID and address proof, bank account details, credit card details, etc.). However, the tenant may be a hacker attacking neighbouring VMs through virtual links. The table below presents the sessions encountered by the tenant LAN. It may be observed that all the sessions initiated from the tenant LAN are with tenant Metadata only through all the virtual machines.

Similarly, the sessions are between tenant Metadata and tenant Vault only for the same virtual machines.

Table 7: The Client DB sessions on Tenants' LAN

Type: LAN
TENAT_LAN1
Client DB Query
- Response Time (sec) <VM1 / TENANT_META>
- Response Time (sec) <VM2 / TENANT_META>
- Response Time (sec) <VM3 / TENANT_META>
- Traffic Received (bytes/sec) <VM1/ TENANT_META>
- Traffic Received (bytes/sec) <VM2/ TENANT_META>
- Traffic Received (bytes/sec) <VM3/ TENANT_META>
- Traffic Sent (bytes/sec) <VM1/ TENANT_META>
- Traffic Sent (bytes/sec) <VM2/ TENANT_META>
- Traffic Sent (bytes/sec) <VM3/ TENANT_META>
- Transaction Size (bytes) <VM1/ TENANT_META>
- Transaction Size (bytes) <VM2/ TENANT_META>
- Transaction Size (bytes) <VM3/ TENANT_META>
IP
IP Processor
TENAT_LAN2
Client DB Query
- Response Time (sec) <VM4 / TENANT_META>
- Response Time (sec) <VM5 / TENANT_META>
- Response Time (sec) <VM6 / TENANT_META>
- Traffic Received (bytes/sec) <VM4/ TENANT_META>
- Traffic Received (bytes/sec) <VM5/ TENANT_META>
- Traffic Received (bytes/sec) <VM6/ TENANT_META>
- Traffic Sent (bytes/sec) <VM4/ TENANT_META>
- Traffic Sent (bytes/sec) <VM5/ TENANT_META>
- Traffic Sent (bytes/sec) <VM6/ TENANT_META>
- Transaction Size (bytes) <VM4/ TENANT_META>
- Transaction Size (bytes) <VM5/ TENANT_META>

The results indicate that the virtual machines cannot jump a layer given their pre-defined destination preferences. In this way, the session inspections and forwarding/dropping are mandatorily implied on each VM. The VMs assigned to the hackers are kept out of the tenant Metadata and the tenant Vault application profiles. In practice, this scenario may be viewed as the unauthorized users not having any records in the metadata or the vault when trying to access a different domain than allowed to them. It may be observed in Figure 58 that the IP packets from the hackers' LAN dropped after an initial attempt. However, these exploits will be detected by the IPS and Anti-malware controls. Hence, the attacker may fail to steal any data from DB_{META} and DB_{VAULT} in spite of gaining access to Cloud VMs by

buying subscriptions. If this is the case, the IPS will check for traces of exploit signatures in the ongoing sessions, and it also will detect the traces of exploit codes and block the session. In addition, anti-malware will check for viruses and spyware signatures in the ongoing sessions. The sessions passing through the IPS may have spyware, adware, or worms embedded in the packets for executing at the BI application layer. Anti-malware will be able to detect and block the session if such malware traces are detected. Hence, traditional security controls are needed on the cloud as well. To get through all the layers and establish unauthorized sessions with the Cloud apps, the hackers will need to break the metadata layer, the vaults layer, the IPS layer, and the antimalware layer. It is unlikely that the hackers will be able to break so many security/privacy layers to reach the Cloud BI applications.

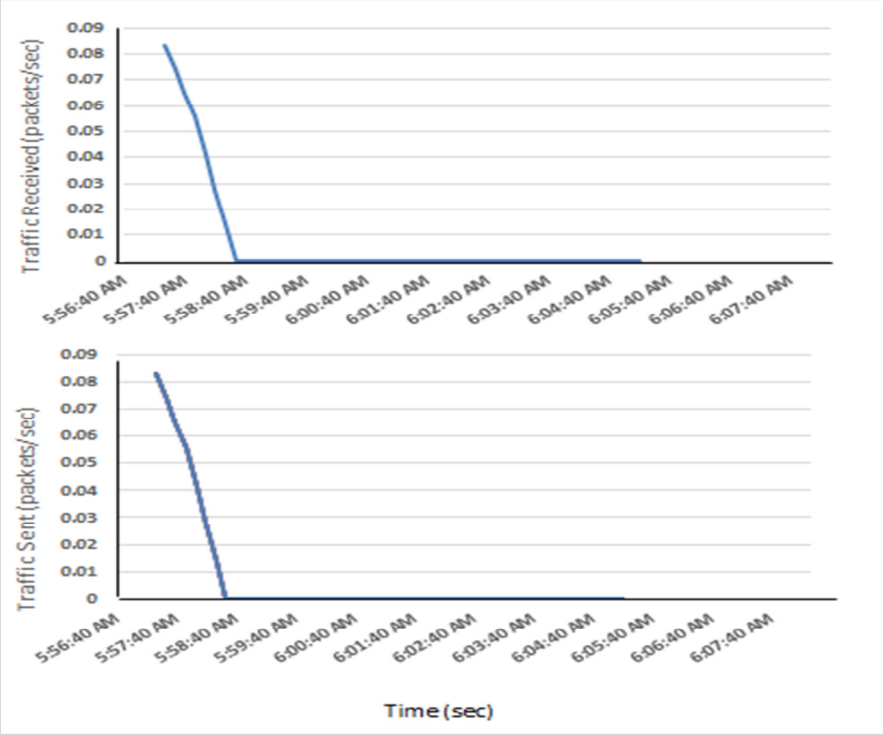

Figure 58: After initial attempts, IP packets from the hackers' machines are dropped

On the other hand, the authorized tenants' LANs could run DB sessions throughout the simulation period as shown in Figure 59. This is because their VMs are added to the

application profiles of all the layers. OPNET modeller does not have the feature to create database tables and enter content in databases. Hence, the simulations are limited to studying their operating behaviour, performance levels, and blocking of attacks.

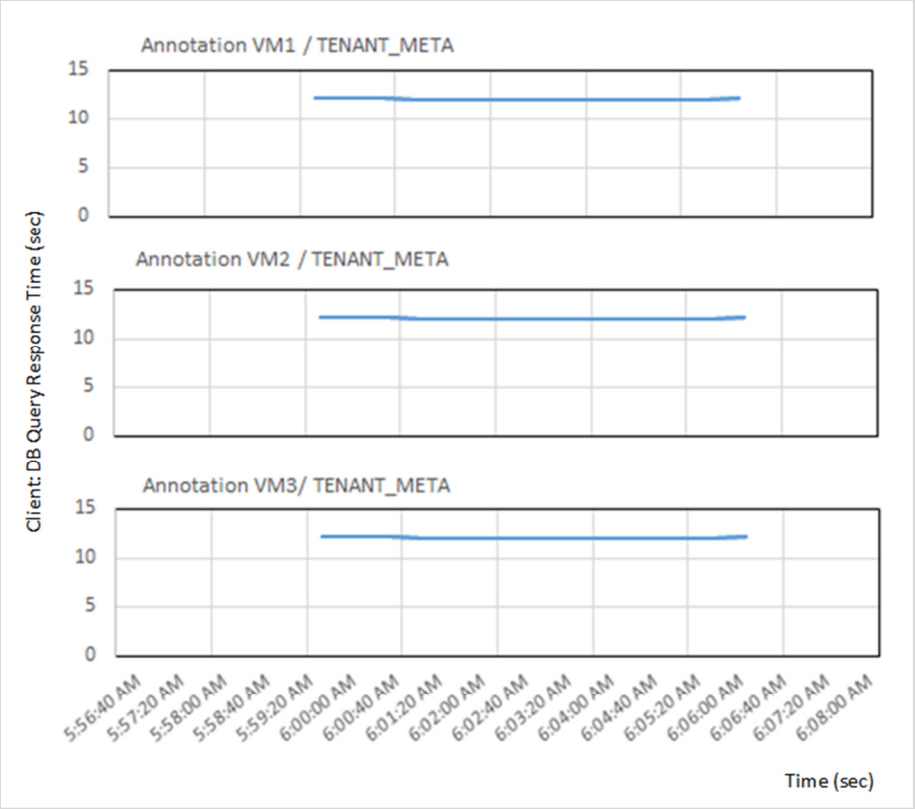

Figure 59: Authorized tenant LANs established and ran DB sessions

The author have used signature based because it is easy to implement, fast, effective and can be applied to protect the whole network without being installed on every participating machine, if one device finds a signature in a process, it can block the process from being passed on to other devices regardless of whether they could protect themselves. This was implemented as a proof of concept and to offer some protection, this is an item that is open for further development for a production system.

One of the issues with cloud computing service access is latency. According to Modi et al (2013), a cloud service provider experiences this delay in response to a client request for a service from the cloud. The “latency” issue could be more significant for Cloud-based business processes which contain collaborating services and users from multiple Cloud systems in different security realms. A typical solution will be locating some intermediate realms that will connect a pair of separate realms, hence serving as an authentication path for collaborating the pair of BI services. Nevertheless the “latency” resulted from creating the authentication path for the two BI services located in disjoint realms is significant as it may involve a lot of other processes for credential conversion that will need extensive invocations to intermediate services. In contrast, in our proposed system, we can authenticate dynamically and enable secure collaboration of different security realms. Hence, the proposed system reduces the latency times that occur due to initiating communication required among a multiple security realms.

For security reasons, if an organization decides moving only part of the BI layers to a Public Cloud. For example, the organization may decide to keep the data in its private cloud datacenter on its premises while using BI tools in Public Cloud environments because they does not trust its sensitive data to a cloud computing provider. Our proposed framework provides a solution to address this issue, as our approach is designed for multiparty authentication for heterogeneous Cloud systems which include both public and private Clouds. For future work, we are planning to carry out experiments in the real system to evaluate the performance and overhead of our framework in this specific scenario.

BI on the Cloud requires multi-layered security for ensuring appropriate protection of each BI layer. The key security layers for BI on the Cloud are lightweight directory access protocol (LDAP), intrusion detection and prevention (IDPS), Antispam, web services

security, antimalware, and database activity monitoring. The key security services for BI on Cloud are identification, authentication, authorization, auditing, data confidentiality, data integrity, data availability, and prevention from unauthorized DB querying and transactions. Hence, traditional security controls are needed on the cloud as well.

Clouds are like galaxies of servers. Multiparty authentication protocol is a proposed mechanism for creating a framework of BI applications used by users having memberships of different security realms but collaborating for running a common project. For example, users from multiple companies sitting on multiple clouds may collaborate for running a common research project with the project manager acting as the trusted principal. The SAC can guarantee secured access to resources on different clouds. However, it cannot guarantee performance and committed response times given that each member cloud may have its own network configurations. Hence, while such an authentication protocol is essential for multiparty collaborations on multiple clouds, timeout settings may not be feasible for all the phases of the authentication protocol. To ensure that the requesting user does not wait for long periods, a localised timeout can be set at the requesting cloud (in this case, it is F). The response time of 60 seconds is a committed performance for executing all the phases, however given that the authentication protocol invokes a complex sequence of interactions throughout the multi-cloud framework and it needs to execute only once for establishing the session, it may be feasible.

5.8.2 Empirical Evaluation

Besides analytic assessments, the feasibility of the proposed authentication system in real-world applications also needs to be examined. Consequently, a series of experiments has been conducted using two types of experimental systems. The first type of experimental system (ES1) consists of a SAC and three experimental services. In ES1 a user creates a

session including three services and then the instances of the three services are included in the session.

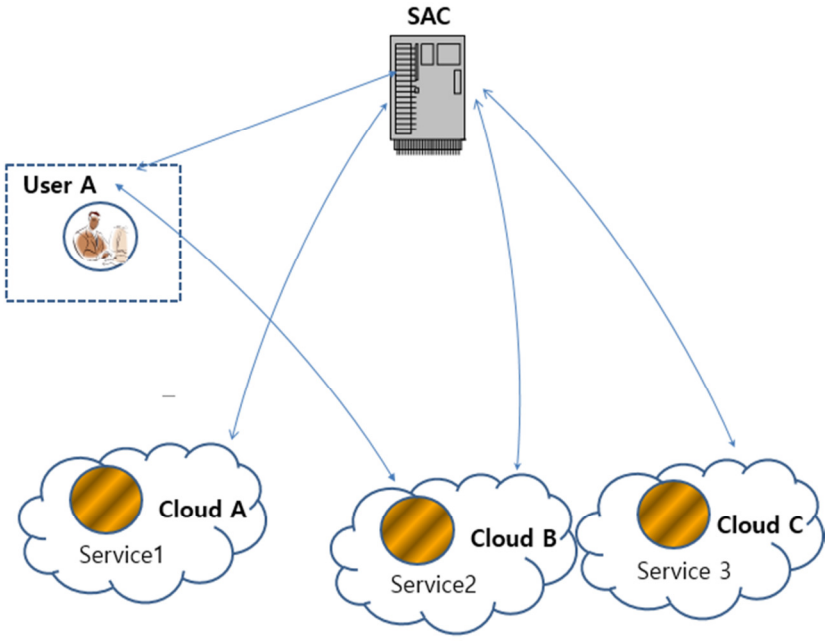

Figure 60: Multi-party authentication system on cloud

The second type of system (ES2) only consists of three experimental services. The experimental services of ES2 will be connected with session users respectively.

ES1 is used to simulate distributed applications with our proposed system while ES2 simulates distributed applications without our proposed system. The overheads of our proposed system can be assessed by comparing the experimental results from these two types of experimental system. We have performed all the experiments evaluated in this section in OPNET modeller.

Figure 61 illustrates the results from the ES1. The number of the session users in different sessions range from 200 to 2000. As illustrated in Figure below, the time consumption of accepting new session users into a session is proportional to the number of the session members.

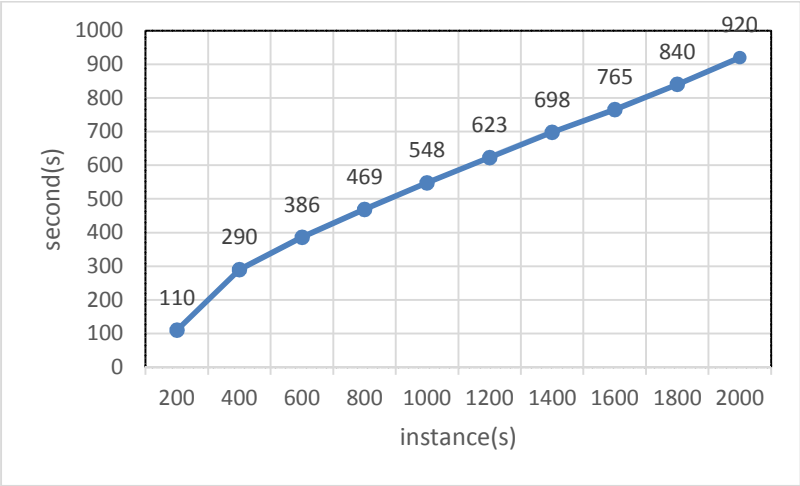

Figure 61: Scalability of the Multi-party Authentication System

In order to extensively evaluate the performances of the proposed system, several other experiments have been implemented. Figures below present the time consumption of invoking database and web services in ES1 and ES2. In these experiments there are no TCP delays, TCP segment delays and TCP retransmission.

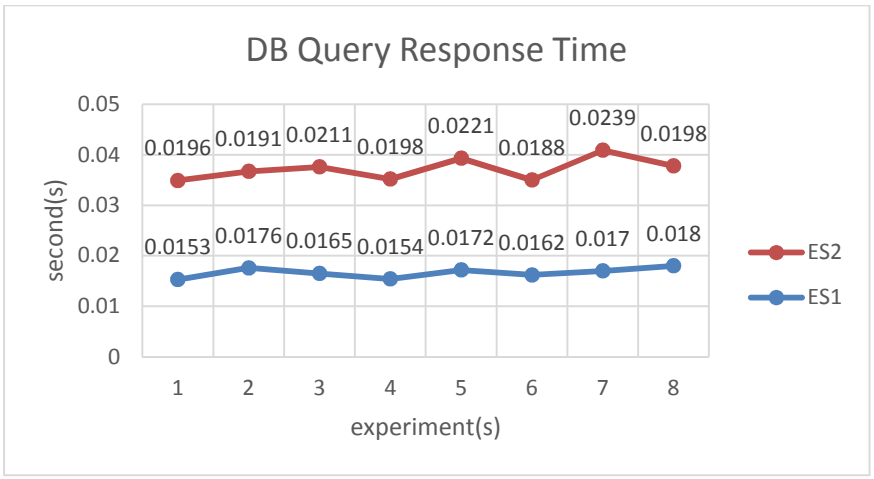

Figure 62: Database query response time

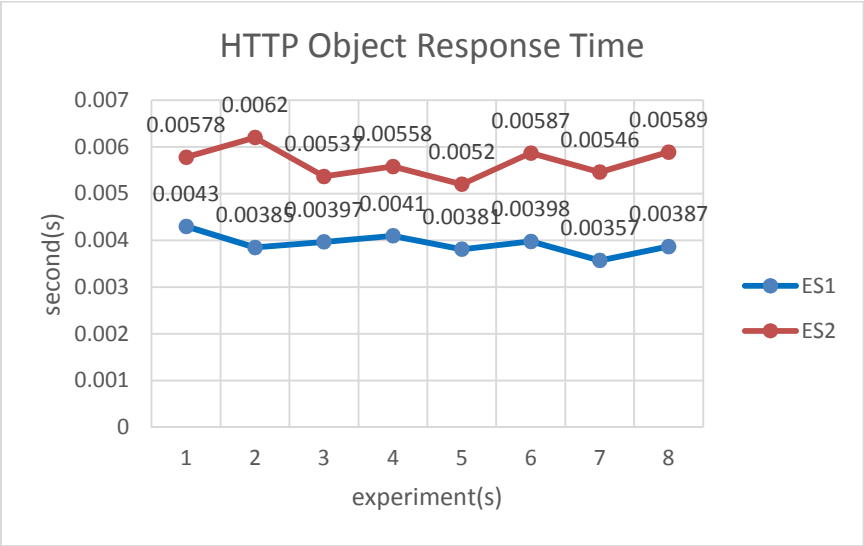

Figure 63: HTTP object response time

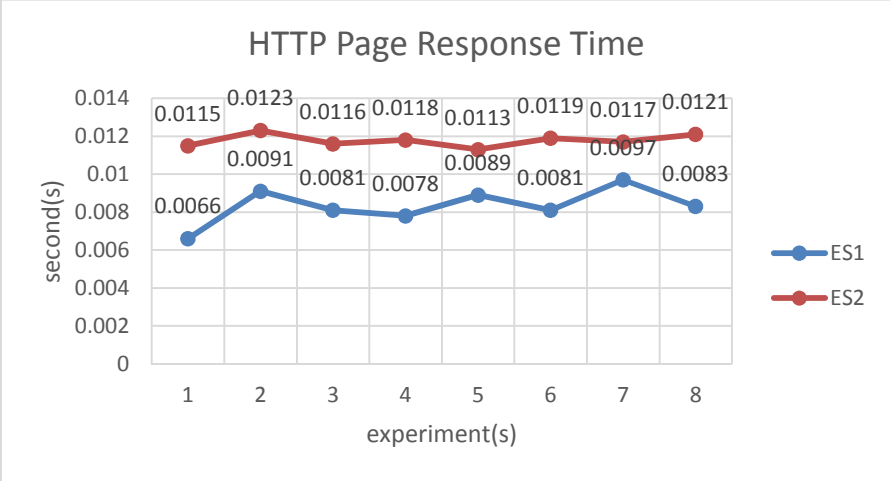

Figure 64: Performance comparison of ES1 and ES2

As illustrated in Figure 64, in the same environments, in ES1 the time consumption of invoking services is less than ES2. It observed that performance of database query and HTML page and object responses in ES1 are much better than ES2.

Now, apart from ES1 and ES2, we also implemented a system to evaluate the performances of Hada and Maruyama's authentication system and Zhang and J. Xu'

authentication system. The time consumption of our proposed solution is better than other solutions. Figure 65 shows the performance comparison of three solutions: Hada and Maruyama's solution, Zhang and J. Xu' solution, and our solution.

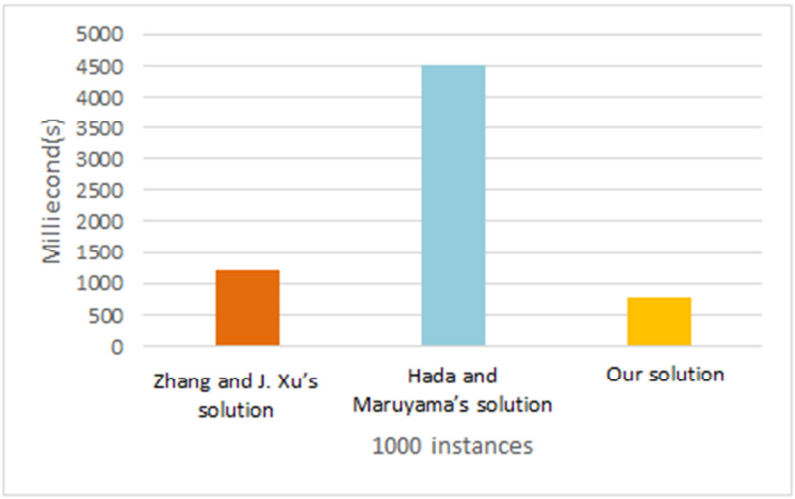

Figure 65: Performance comparison

As Figure 65 illustrated, the time consumption of in our solution is significantly less than other multi-party authentication solutions. The number of the session members (users and services) that the experimental systems can support is influenced by several factors.

A critical factor is the security protocol which the authentication system is combined with. When used with different security protocols, the performance of ES1 varies noticably.

For example, when ES1 is deployed and uses the security protocol, the time consumption increases nonlinearly after over 940 instances are generated. When ES1 is combined with the secure message protocol, the system can execute stably even after generating more than 10,000 instances. Another critical factor is the memory space. This also affects the performance of an experimental system. Information of sessions, session members and experimental services need to be stored in memory. As the amount of such data, the free

space of physical memory decreases. For this reason the operating system detects lack of physical memory and frequently attempts memory swap. Thus the performance will reduce.

But when an experimental system is deployed in a distributed environment, the SAC and the experimental services do not need to utilize the same limited memory. The performance of experimental systems therefore becomes much better.

The experiments introduced in this chapter were conducted using OPNET modeller to assess the scalability of the Multi-party authentication system for BI on cloud, the compatibility of the system with other message-level security protocols, and the runtime overheads of the mechanism under different conditions. The experimental results show that the time consumption increases linearly as the number of session members increases. Our solution is therefore potentially applicable for Cloud computing with a large number of participants. Various public key algorithms are also compared and evaluated during the experiments in order to select the most suitable one for our new protocol.

In addition, the overheads of the Multi-party authentication system for BI on cloud are comparable with other standard multi-party solutions. In the experiments deployed in a distributed environment, the experimental services collaborate over a local area network. The time consumption of experimental systems is very small, and most of the runtime overhead is imposed by the cryptographic key generation. In such environments, the performance of the proposed system can be improved by employing effective cryptographic algorithms. However, in a real-world application, session members may be geographically distributed and communicate over the Internet. In these applications, the time consumption can be much larger than that in our experimental systems. For example, in some experiments, it may take several seconds to transfer a message between countries which are geographically remote from each other. In such cases, the performance of distributed

applications is largely affected by the geographic location of session members and the broadband of networks, and it is a reasonable method to improve the performance of the proposed authentication system on cloud by reducing additional messages introduced by authentication processes.

When compared a final model with the benchmark model, the experiment was carried out with the same environment in terms of hardware specifications and bandwidths. So that we can confirm that the improvement of performance is resulted from our proposed security protocol/architecture it is not because hardware.

5.8.3 Validation Results

The validation of results is an essential part of the research process. The author has validated the results in two ways. First, the correctness of framework has been verified theoretically by using the well-known BAN logic to analyse authentication protocols by deriving the beliefs that honest principals in protocol. Second, the framework was practically implemented through developing a prototype system using Eclipse. The prototype system was developed to implement the five multiparty authentication protocols in the multiparty authentication system. In addition, the performance of the proposed framework was evaluated by comparing with two state-of-the-art methods. The comparison results show that our proposed method achieved better performance than these two state-of-the-art methods.

5.9 **Risk Assessment**

An assessment of risk is about acknowledging that risks may occur during the research work, whether enough precautions have been taken and whether more precautions are necessary to prevent risk (Ashworth, 2012). Risk assessment is very important and should be completed in any research project because it is essential to keep data secure and ensure that critical data are protected against loss through hard-drive failure, corruption or

equipment damage. Another potential risk for this project is that the work is not completed on time. In order avoid risk during a project, the author ensured that data were backed up securely at the end of each day; kept copies of data on multiple secure storage devices; used an anti-virus software and keep it updated to protect the computer against malicious computer viruses, Trojans and other mal-ware; and built in extra time at key stages of the project and monitor progress against the project's Gantt chart. Risk assessment was carried out at the beginning of my PhD project. Risk assessment has also been approved by the College Research Committee in my RD5 and RD7 reports. Due to immense care and consideration, none of these risks happened during my PhD.

5.10 **Ethical and Legal Issues**

In recent years, ethical and legal issues have come to the forefront of many scientists' attention. Ethical and legal concerns are very important and should be considered in any research project because they may help us decide whether the research should even be done, and if so, how it should be pursued (Davenport, 2008). As the importance of ethical scrutiny of research projects has grown, formal ethical approval must take place before PhD researchers are allowed to conduct research projects. Ethical approval has been secured by the College Research Committee in my RD5 and RD7 reports prior to the commencement of fieldwork. There are no ethical or legal issues arose during the work presented in the PhD project, and there is no involvement of either human beings or animals. Any companies that want to implement my framework also need to fully consider ethical and legal issues.

5.11 **Summary**

In this chapter, the multiparty authentication system for BI and the mechanisms for hosting it on cloud computing were discussed and five core protocols of our multiparty authentication system have been presented. The correctness of the protocol is formally

analysed and proven using the well-known BAN logic. A comprehensive empirical study is performed to evaluate the scalability and the runtime overhead of the authentication system. The author address this problem by designing and implementing a new multiparty authentication protocol for dynamic authentication interactions when members of different security realms want to access distributed BI services through a trusted principal. This mechanism can help Cloud session users authenticate their session membership so as to largely simplify the authentication processes within multi-party sessions.

Finally in this section, risks, ethics and legal implications are considered including mitigation factors and recommendations.

Chapter 6: Conclusions and Future Directions

6.1 **Introduction**

This chapter discusses the major contribution of the thesis, conclusions and future directions of this study. The experimental analysis highlights the contribution and the productive results. In addition, this section highlights limitations of this work that have been identified as the work progressed. The limitations are not failings, but do provide scope and opportunity for further research and development.

6.2 **Major Contributions of the Study**

This research has considered problems associated with reliable, timely and secure data transfer mechanisms necessary for shared Business Intelligence data processing networks. The stated aim was to develop a multi-party authentication model for securing business intelligence on cloud computing. This aim has been achieved through the definition of a multi-party authentication system framework. Suitability and robustness of the framework has been shown through simulation and experimentation.

Overall, the most significant contribution has been to propose a system and procedure which simplifies the authentication process which is undertaken between two unrelated secure business environments prior to allowing data interchange. The simplification improves the speed of authentication, thereby reducing overall transaction time and is achieved without compromising the security of either party.

- Major components of the system include:
  - A unique set of management and authentication protocols
  - A cloud based session authority

The set of protocols that were designed specifically to enable multi-party session management and cross-realm authentication were implemented to satisfy the following list of requirements:

- Address scenarios of Business Intelligence (BI) application access on cloud platforms when members of different security realms need to access distributed BI services through a trusted principal.
- Help session users authenticate their session memberships to simplify the authentication process in multiparty sessions without reducing security.
- Support fine-grained session key management where session cloud users join and leave the session frequently.
- Prevent the service instances in a multi-party session from communicating outside of the session; the multi-party session authentication mechanism therefore partially achieves the functionality of a protocol firewall.

The session authority cloud (SAC):

- Control sessions in the clouds,
- Authorise sessions based on matching keys which are forwarded by multiparty session handler.
- Serve as the cloud certification authority.

6.3 **Conclusions**

Cloud is an important part of future BI and offers several advantages in terms of cost efficiency, reliability, flexibility and scalability of implementation. Moreover, it offers the opportunity to take advantage of enhanced data sharing capabilities. Cloud has the potential to offer a new lease of life to BI and OLAP frameworks. Cloud computing comprises three primary methods of provisioning services – software-as-a-service (SaaS), platform-as-a-

service (PaaS) and infrastructure-as-a-service (IaaS). These complementary but contrasting services may be provided by the same or different providers depending upon the business arrangements. However, regardless of the contractual arrangements, the SaaS provider needs the settings on the PaaS and IaaS Clouds to be defined as per the application services provided through the web services architecture components. Cloud provision includes the service provisioning and routing engines that can effectively sense the loading pattern on the underlying resources to determine the requirement to undertake migration.

The security risks and solutions for cloud computing environments have been presented; the NIST recommendations on managing risks on cloud computing are reviewed and analysed in detail. More specific to the purpose of this research, the techniques of deploying Business Intelligence on cloud computing platforms and the security threats facing business intelligence have been discussed. Furthermore, the modelling scenarios related to cloud security, BI on the cloud, and BI security on the cloud have been presented and critically reviewed.

In this research, the lower level modelling activity helped in preparing the fundamentals for the definitive model to be developed as the final outcome of this research. The author have proposed a new authentication mechanism model for multi-party authentication for BI, mechanisms for hosting on cloud computing and five core protocols all of which combine to form the multiparty authentication system.

This multiparty authentication system for dynamic authentication interactions is effective when members of different security realms want to access distributed BI services through a trusted principal. This mechanism can help Cloud session users authenticate their session membership so as to largely simplify the authentication processes within multi-party sessions. In addition, there is an improvement in the performance of the proposed authentication system on cloud due to a reduction in the additional messages introduced by

the authentication processes. The correctness of the protocol is formally analysed and proven using the well-known BAN logic.

To achieve these goals, the authentication protocols are designed in the following way:

- Assign each session user a Diffie-Hellman key pair. The public key is used as the identifier of the instance while the key is kept privately. All such sessions will be identified by the SAC.
- Assign a session authority cloud to control sessions in the clouds, and to authorise the sessions based on matching of keys forwarded by multiparty session handler. It also serves as the cloud certification authority. In addition, to maintain the status information about session cloud users.
- Allow each pair of session cloud user to verify each other's identity under the assistance of the Session Authority Cloud and then generate a shared secret key for exchanging information confidentially to secure their communication.

The multi-party authentication mechanism can prevent the service instances in a particular multi-party session from exchanging messages with the outside so that the multi-party session authentication mechanism can partially achieve the functionality of the firewalls. This mechanism solution supports fine-grained session key management where session cloud users join and leave the session frequently. Also, with our solution the management of session key can be more readily facilitated. Finally, empirical evaluation is carried out to assess the scalability and the runtime overhead of the authentication system. The objectives of the multiparty authentication protocols are proven to achieve the intended goals theoretically and experimentally. The experimental results clearly indicate that the proposed protocol and its implementation have a sound level of scalability and impose only a

modest degree of performance overhead which is comparable with other security related overheads.

6.4 **Future Research Directions**

The research aim and objectives have been achieved; scope remains for further interesting development of the multiparty authentication system. Although in our authentication mechanism the amount of communications between the cloud users of a session and the session cloud authority is limited, there is potential for the performance overheads to cause practical concern in a practical network. Greater functionality and reporting could be achieved if the SAC communication message content were to be extended; however, there would be a probable increase in communication overhead would also need to be assessed and addressed if necessary.

While this research has presented the framework, additional research and development are needed to implement a practical hybrid model of BI security on the cloud. The application of XML data structures could be investigated as a means to standardise and improve communication efficiency and assist with developing a full production implementation. Research into the continually developing state-of-the-art methodologies which pertain to service provisioning, service routing, schema partitioning and load balancing would all be beneficial for implementing an enterprise level RDBMS system with the aim of achieving a massively parallel processing RDBMS server system for taking BI to the Clouds.

Cloud computing offers significant computing power and capacity. Hence, BI is expected to enter many complex domains (business and non-business related) which were impossible for it in a self-hosted environment. Applications like context-awareness, location-aware automation, massive scale semantics, advanced science and technology databases, real time disaster and crisis management, city management, global finance and economy reporting

and the global monitoring of industries and sectors are just a few of the areas where BI or BI like systems offer tremendous potential when combined with Cloud computing. The size, scale, dynamism and scope of data marts and data warehouses on Clouds may exceed even the Petabytes scale (one of the the emerging challenges of Big Data). Such data systems cannot be managed using traditional systems and tools. The security challenges at such massive scales will be different and much more complex. Hence, it is imperative to establish ways to increase the security of the cryptosystem without compromising the temporal functionality, which are foundation elements to our solution.

Reference List

Abadi, D. J. (2009). "Data Management in the cloud: Limitations and Opportunities", *Data Engineering*, Vol. 32 (1): 3-11, IEEE Computer Society.

Aboelela, E. (2003). "Network Simulation Experiments Manual". Morgan Kauffmann: 2-104.

Aboulnaga, A., Salem, K., Soror, A. A., Minhas, U. F., Kokoseilis, P. and Kamath, S. S. (2009). "Deploying database appliances on the cloud". *Data Engineering*, Vol. 32 (1): pp. 13-20. IEEE Computer Society.

Agrawal, R., Srikant, R., and Thomas, D. (2005). "Privacy Preserving OLAP", In: *SIGMOD 2005*, June 14-16, 2005, Baltimore, Maryland, USA, 1-12, ACM.

Ahmad, S. and Ahmad, R. (2010). "An Improved Security Framework for Data Warehouse: A Hybrid Approach", *IEEE Computer Society*: 1586-1590. IEEE.

Al-Aqrabi, H., Liu, L., Hill, R., and Antonopoulos, N. (2014). "Cloud BI: Future of business intelligence in the Cloud", Journal of Computer System Science, Elsevier.

Al-Aqrabi, H. and L Liu. IT Security and Governance Compliant Service Oriented Computing in Cloud Computing, Book Chapter for Principles, Methodologies, and Service-Oriented Approaches for Cloud Computing, in press, January, 2013, IGI Global, USA.

Al-Aqrabi, H., Liu, L., Hill, R., and Antonopoulos, N. (2012). "Taking the Business Intelligence to the clouds". Proceedings of 14thIEEE International Symposium on (HPCC2012), Liverpool, UK, June 25-27, 2012.

Al-Aqrabi, H et al. (2012). “Investigation of IT Security and Compliance Challenges in Security-as-a Service for Cloud Computing", Proceedings of 15thIEEE International Symposium on (ISORC2012), Shenzhen, China, April 11-13, 2012.

.Al-Aqrabi, H et al. 2013. "Business Intelligence Security on the Cloud: Challenges, Solutions and Future Directions", Proceedings of 7th International Symposium on (SOSE2013), San Francisco Bay, USA, March 25 - 28, 2013.

Al-Aqrabi, H., Liu, L., Hill, R., and Antonopoulos, N. (2014). "A Multi-layer Hierarchical Inter-Cloud Connectivity Model for Sequential Packet Inspection of Tenant Sessions Accessing BI as a Service" (HPCC2014). 20-22 Aug. 2014, Paris, France.

Alborz, N., Keyvani, M., Nikolic, M. and Trajkovic, L. (2000). “Simulation of Packet Data Networks using OPNET”. *ACM*: 1-6.

Amburst, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I. and Zaharia, M. (2009). “Above the Clouds: A Berkeley View of Cloud Computing”. Technical Report No. UCB/EECS-2009-28, Electrical Engineering and Computer Sciences, University of California at Berkeley, p.2. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf [Accessed: 24 January, 2012]

Amburst, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. (2010). “A View of Cloud Computing”. *Communications of the ACM*, Vol. 53 (4): p. 50-58. ACM.

Aulbach, S. Grust, T,. Jacobs, D., Kemper, A. and Rittinger, J. (2008). “Multi-tenant databases for Software as a Service: Schema Mapping Techniques”. *ACM* Transactions, p. 1195-1206.

Ashworth, A. Contractual procedures in the construction industry. 6th ed. Harlow: Pearson, 2012.

Avizienis, A. Laprie, J.-C., Randell, B. & Landwehr, C. (2004), "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, p. 11-33.

Badger, L., Bohn, R., Chu, S., Hogan, M., Liu, F., Kaufmann, V., Mao, J., Messina, J., Mills, K., Sokol, A., Tong, J., Whiteside, F. and Leaf, D. (2011). “U.S. Government cloud computing technology roadmap – Volume II”, Special Publication 500-293, NIST (U.S. Department of Commerce), p. 10-15.

Bento, A. and Bento, R. (2011). " Cloud Computing: A new phase in IT Management ". *Journal of Information Technology Management*, 22 (1): 39-46. Elsevier.

Blanco, C., Pérez-Castillo, R., Hernández, A., Fernández-Medina, E., and Trujillo, J. (2009). "Towards a Modernization Process for Secure Data Warehouses", T.B. Pedersen, M.K. Mohania, and A M. Tjoa (Eds.), *LNCS 5691*: 24–35, Springer-Verlag Berlin Heidelberg.

Bisong, A. and Rahman, S. M. (2011). "An Overview of the security concerns in enterprise cloud computing". *International Journal of Network Security and its Applications (IJNSA)*, Vol.3 (1), p. 30-45.

Bolze, R. and Deelman, E. (2011). “Exploiting the Cloud of Computing Environments: An Application’s Perspective”. p. 173-196. Book Chapter: *Cloud Computing and Software Services: Theory and Techniques*. Ahson, S. A. and Ilyas, M. (Eds). FL: CRC Press, Taylor and Francis Group.

Boutsinas, B. (2005). “On defining OLAP formulations”. *IMA Journal of Management Mathematics*, Vol. 16: pp. 339–354. ABI Informs.

Brakmo, L. S. and Peterson, L. L. (1996). “Experiences with Network Simulation”. In *proceedings of ACM SIGMETRICS ’96 International Conference on Measurement and Modeling of Computer Systems*, 1-16. ACM Digital Library.

Brankovic, L. and Estivill-Castro, V. (2000). "Privacy issues in knowledge discovery and data mining", Department of Computer Science and Software Engineering, The University of Newcastle, Australia, 1-12.

Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidmann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y. and Yu, H. (2000). “Advances in Network Simulation”. IEEE: 59-66.

Brieter, G. (2010). "Cloud computing architecture and strategy". *IBM Blue Books*: 3-4, IBM Research.

Carroll, M., Merwe, A. and Kotze, P. (2011). "Secure Cloud Computing: Benefits, Risks and Controls". *IEEE*, p. 1-9.

Carvalho, M. (2011). “SECaaS–Security as a Service”. *ISSA Journal*, p. 20-24.

Chadha, B. and Iyer, M. (2010). “BI in the cloud”. *SET-Lab briefings*, Vol. 8 (1): pp. 39-44. Infosys Research.

Chang, X. (1999). “Network Simulations with OPNET”. IEEE: 307-314.

Chao, Y., Bingyao, C., Jiaying, D. and Wei, G. (2010). “The Research and Implementation of UTM”. *IEEE*, p. 389-392.

Chea, J., Duanb, Y., Zhanga, T, and Fana, J. (2011). "Study on the security models and strategies of cloud computing", *Procedia Engineering*, Vol. 23: 586 – 593, SciVerse, Science Direct.

Che, J., Duan, Y., Zhang, T., and Fan, J. (2011). "Study on the security models and strategies of cloud computing", *Procedia Engineering*, Vol. 23: 586–593, Elsevier.

Chorafas, D. N. (2011). “Cloud Computing Strategies”. p. 65-70. London: CRC Press. Taylor and Francis Group.

Convery, N. (2010) "Cloud computing toolkit: Guidance for outsourcing information storage to the cloud". Department of Information Studies, Aberystwyth University. UK and Ireland: Archives and Records Association, 4-70.

Cottrell, R. L., Logg, C., Chhaparia, M,. Grigoriev, M., Haro, F., Nazir, F. and Sandford, M. (2006). “Evaluation of Techniques to Detect Significant Network Performance Problems using End-to-End Active Network Measurements”. In *Proceedings of the IEEE/IFIP Network Operations & Management Symposium (NOMS 2006)*, April 3-7, 2006, Vancouver, Canada. IEEE: 2-11.

Curino, C, Jones, E. P. C., Popa, R. A., Malvia, N., Wu, E., Madden, S., Balakrishnan, H. and Zeldovich, N. (2011). “Relational Cloud: A Database-as-a-Service for the Cloud”. In *5th Biennial Conference on Innovative Data Systems Research (CIDR 2011),* January 9-12, 2011 Asilomar, California. 235-240. Massachusetts Institute of Technology.

Cuzzocrea, A., Bertino, E., and Sacca, D. (2012). "Towards a Theory for Privacy Preserving Distributed OLAP", In: *PAIS’12*, March 30, 2012, Berlin, Germany, pp. 1-6, ACM.

Cuzzocrea A., Sacca, D. and Serafino, P. (2007). “Semantics-aware advanced OLAP visualisation of multi-dimensional data cubes”. *International Journal of Data Warehousing and Mining*, Vol. 3 (4): 1-30. IGI Publishing.

Celesti, A., Tusa, F., Villari, M., & Puliafito, A. (2010b), "Security and Cloud Computing: InterCloud Identity Management Infrastructure", *IEEE Computer Society*, p. 263-265.

Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R 2011, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms". Software: Practice and Experience 41 (1): 23–50.

Chen, C. & Tu, J. (2013), "A Novel Cloud Computing Algorithm of Security and Privacy", *Mathematical Problems in Engineering*, Volume 2013: p. 1-6, Hindawi Publishing Corporation.

Conradi, R., Wang, A. I., & Esernet. (2003). *Empirical methods and studies in software engineering: Experiences from ESERNET*. Berlin: Springer.

Cremers, C., & Mauw, S. (2012). *Operational semantics and verification of security protocols*. Berlin: Springer.

Dai, Q., Zhao, X., Xu, Q., & Jiang, H. (2011), "A New Cross-realm Group Password-based Authenticated Key Exchange Protocol", *IEEE Computer Society*, p. 856-860.

Dash, D., Rao., J., Megiddo, N., Ailamaki, A., and Lohman, G. (2008). “Dynamic Faceted Search for Discovery-driven Analysis”. In *CIKM’ 08*, October 26–30, 2008, Napa Valley, California, USA. 1-10. ACM.

Davenport, L.D. 2008. A sense making approach to ethics training for scientists: Preliminary evidence of training effectiveness. Ethics and Behavior, 18(4): 315–339. Taylor & Francis.

Demchenko, Y. and Laat, C. D. (2011). “Defining Generic Architecture for Cloud Infrastructure as a Service Model”. In *The International Symposium on Grids and Clouds and the Open Grid Forum Academia Sinica*, March 19 - 25, 2011, Taipei, Taiwan, ACM: 2-10.

Draxler, S., Stevens, G., and Boden, A. (2015): Keeping the development environment up to date - A Study of the Situated Practices of Appropriating the Eclipse IDE, in: IEEE Transactions on Software Engineering

Ekanayake, J. Qiu, X. Gunarathne, T, Beason, S. and Fox, G. (2011). “High-performance parallel computing with cloud and cloud technologies”. Book Chapter: *Cloud Computing and Software Services: Theory and Techniques*” Ahson, S. A. and Ilyas, M. (Eds). FL: CRC Press, Taylor and Francis Group, 276-307.

Farhan, M. S., Marie, M. E., El-Fangary, L. M., and Helmy, Y. K. (2012). "Transforming Conceptual Model into Logical Model for Temporal Data Warehouse Security: A Case Study", *International Journal of Advanced Computer Science and Applications*, Vol. 3 (3): 115-122.

Fernandez-Medinaa, E., Trujillo, J., Villarroel, R., Piattini, M. (2007). "Developing secure data warehouses with a UML extension", *Information Systems*, Vol. 32: 826–856. Elsevier.

Fienberg, S. E. (2006). "Privacy and Confidentiality in an e-Commerce World: Data Mining, Data Warehousing, Matching and Disclosure Limitation", *Statistical Science*, Vol. 21 (2): 143–154, Institute of mathematical Statistics.

Giovinazzo, W. A. (2002). "Internet-enabled Business Intelligence", NY: Prentice-Hall.

Glaser, J. and Stone, J. (2008) "Effective use of Business Intelligence". *Healthcare Financial Management*, Vol. 62 (2): 68-72. ABI/INFORM Global.

Golfarelli, M., Maio, D., and Rizzi, S. (1998). "Conceptual Design of Data Warehouses from E/R Schemes", *In the Proceedings of the Hawaii International Conference on System Sciences*, January 6-9, 1998, Kona, Hawaii. IEEE: pp. 1-10.

Gowrigolla, B., Sivaji, S. and Masillamani, M. R. (2010). "Design and Auditing of Cloud Computing Security". *IEEE*: 292-297.

Grabova, O., Darmont, J., Chauchat, J., and Zolotaryova, I. (2011). "Business Intelligence for Small and Middle-Sized Entreprises", University of Lyon: 1-12.

Grimes, S. (2006). “New Directions for OLAP”. *Intelligent Enterprise*, Vol. 9 (3): 10. Pro-Quest Computing.

Gross, J., Günes, M., & Wehrle, K. (2010). *Modeling and Tools for Network Simulation*. Berlin: Springer Berlin.Gul, I., Rehman, A. U. and Islam, M. H. (2011). “Cloud Computing Security Auditing”, *IEEE*, p. 143-148.

Guo, J., Xiang, W. and Wang, S. (2007). “Reinforce Networking Theory with OPNET Simulation”. *Journal of Information Technology Education*, Vol. 6: 215-223. University of Michigan, Dearborn.

Han, S. and Xing, J. (2011). "Ensuring data storage security through a novel third party auditor scheme in cloud computing", *IEEE*, 264-268.

Huang, C. and Tseng, T. (2009). "Analytical Knowledge Warehousing for Business Intelligence", Book Chapter: *Encyclopedia of Data Warehousing and Mining*, John Wang (Ed), 2nd Edition, Vol. I, IGI Publishing: p. 31-38.

Hummer, W., Bauer, A., Harde, G. (2003). “XCube – XML for Data Warehouses”. *ACM Transactions*: 33-40. ACM.

Hada, S and Maruyama, H. (2002) “Session Authentication Protocol for Web Services,” Proc. Symposium on Application and the Internet, 2002, pp. 158-165.

Iankoulova, I. and Daneva, M., 2012, May. Cloud computing security requirements: A systematic review. In Research Challenges in Information Science (RCIS), 2012 Sixth International Conference on (pp. 1-7). IEEE.

Jansen, W. and Grance, T. (2011). "Guidelines on Security and Privacy in Public Cloud Computing". Special Publication 800-144, p. 4-60. National Institute of Standards and Technology (NIST), U.S. Department of Commerce.

Joung, P. (2003). “General Network Performance Testing Methodology”. *Sprint Communication Labs*: 3-24.

Kadan, A. (2012). "Security Management of Intelligent Technologies in Business Intelligence Systems", Yanka Kupala State University of Grodno: 1-3.

Kashyap, A., Hristidis, V., and Petropoulos, M. (2010). “FACeTOR: Cost-Driven Exploration of Faceted Query Results”. In *CIKM’ 10*, October 25–29, 2010, Toronto, Ontario, Canada. 1-10. ACM.

Katic, N., Quirchmayr, G., Schiefer, J., Stolba, M., and Tjoa, A. M. (2006). "A Prototype Model for Data Warehouse Security Based on Metadata", University of Vienna: 1-9.

Katzan, H. Jr. (2010). "On The Privacy Of Cloud Computing", *International Journal of Management and Information Systems*, Vol. 14 (2): 5-12.

Kirkgoze, R., Katic, N., Stolba, M., and Tjoa, A. M. (1997). "A Security concept for OLAP", *IEEE Computer Society*: 619-626. IEEE.

La, H. J. and Kim, S. D. (2009). “A Systematic Process for Developing High Quality SaaS Cloud Services”. LNCS 5931. p. 3-8. Jaatun, M. G., Zhao, G. and Rong, C (Eds). Springer-Verlag Berlin Heidelberg.

Lehner, W. (2007). “Modeling Large Scale OLAP Scenarios”. University of Erlangen-Nuremberg: 1-15.

Lempel, R. and Sheinwald, D. (2010) "Cubing by composition of faceted search". *IBM Research*: 3-28.

Litoiu, M. and Litoiu, M. (2010). "Optimizing Resources in Cloud, a SOA Governance View". *Proceedings of Governance of Technology, Information and Policies: Addressing the Challenges of Worldwide Interconnectivity*, 7 December 2010, Austin

(Texas), USA. The Association for Computing Machinery (ACM) International Conference Proceedings Series. p. 73-74.

Li, H., Dai, Y., Tian, L., & Yang, H. (2009), "Identity-Based Authentication for Cloud Computing", CloudCom 2009, M.G. Jaatun, G. Zhao, and C. Rong (Eds.), *LNCS 5931*, p. 157–166, Springer.

Li, H., Dai, Y., & Yang, B. (2011), "Identity-Based Cryptography for Cloud Security", University of Electronic Science and Technology of China and University of Tennessee, USA, *https://eprint.iacr.org/2011/169.pdf* [Accessed: 24 August, 2014], p. 1-9.

MacDonald, N. (2010). "Securing the Next-Generation Virtualized Data Center". *Gartner Report no. G00173434*: 2-50.

MacVittie, L. (2005). "Business Intelligence – One Suite to serve them all". *Network Computing*, Vol. 16 (21): 31-36. ABI/INFORM Global.

Mahboubi, H., Hachicha, M., and Darmont, J. (2009). "XML Warehousing and OLAP", Book Chapter: *Encyclopedia of Data Warehousing and Mining*, John Wang (Ed), 2nd Edition, Vol. IV, IGI Publishing: p. 2109-2116.

Malinowski, E. and Zimanyi, E. (2008). "Advanced Data Warehouse Design", *Springer-Verlag Berlin Heidelberg*: 1-24.

Modi, C., Patel, D., Borisaniya, B., Patel, A. and Rajarajan, M., 2013. A survey on security issues and solutions at different layers of Cloud computing. The Journal of Supercomputing, 63(2), pp.561-592.

Mietzner, R. and Leymann, F. (2008). “Towards Provisioning the Cloud: On the Usage of Multi-Granularity Flows and Services to Realize a Unified Provisioning Infrastructure for SaaS Applications”. *IEEE Computer Society*, p. 1-8. IEEE.

Miller, M. (2009). “Cloud Computing: Web based applications that change the way you work and collaborate online”. US: Que Publishing (Pearson). p. 24-30.

Mukhin, V. and Volokyata, A. (2011). "Security Risk Analysis for Cloud Computing Systems". *The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, p. 737-742. 15th September 2011 to 17th September 2011, Prague, Czech Republic. IEEE.

M. Burrows, M. Abadi, and R. Needham, “A Logic of Authentication,” ACM Trans. on Computer Systems, vol. 8, no. 1, pp. 18-36, Feb. 1990.

M. Burrows, M. Abadi, and R. Needham, “Authentication: A Practical Study of Belief in Action,” Proc. the Second Conference on Theoretical Aspects of Reasoning About Knowledge (Tark), Mar. 1988, pp. 325-342.

“Network Simulation using OPNET”. Laboratory Manual on Multiservice Communication Networks. School of Engineering and Mathematical Sciences, City University, London, 2006, 3-27.

NIST (2011). “US Government Cloud Computing Technology Roadmap”. Special Publication 500-293. Cloud Computing Program, National Institute of Standards and Technology (NIST), US Department of Commerce, 13-78.

Niyato, D. (2011). "Optimization-Based Virtual Machine Manager for Private Cloud Computing". *IEEE Computer Society*, p. 99-106. IEEE.

Ouf, S. and Nasr, M. (2011). "Business Intelligence in the cloud", *IEEE Computer Society*: 650-655. IEEE.

Ouf, S. and Nasr, M. (2011). "Business Intelligence Software as a Service (SaaS)", *IEEE Computer Society*: 641-649. IEEE.

Ouf, S. and Nasr, M. (2011). "The Cloud Computing: The Future of BI in the Cloud", *International Journal of Computer Theory and Engineering*, Vol. 3 (6): 750-754.

Qin, B., Wang, H., Wu, Q., Liu, J., & Domingo-Ferrer, J. (2013), "Simultaneous authentication and secrecy in identity-based data upload to cloud", *Cluster Computing*, Vol. 16 (4), p. 845-859, Springer.

Pearson, S. and Charlesworth, A. (2009). "Accountability as a Way Forward for Privacy Protection in the Cloud". Proceedings of CloudCom 2009, December 2009, Beijing. p. 3-15. Springer LNCS.

Pearson, S. (2009). “Taking Account of Privacy when Designing Cloud Computing Services”. Produced by HP laboratories for *IEEE*, p. 2-10.

Phelps, J. R. and Dawson, P. (2007). “"Demystifying Server Virtualization Taxonomy and Terminology", Report ID Number: G00148373, p. 1-9. Gartner Research.

Preston, R. (2007). “We've Yet To Even Reach The Wonder Years Of BI”. *Information Week*, January 1 to January 8, 2007, Issue No. 1120: 62-63. ABI/INFORM Global.

Priebe, T. and Pernul, G. (2001). Proceedings of 20th International Conference on Conceptual Modeling (ER 2001), November 27-30, 2001, Yokohama, Japan, 1-14, ACM.

Priebe, T. and Pernul, G. (2000). "Towards OLAP Security Design – Survey and Research Issues", In: *Proceedings of Third ACM International Workshop on Data Warehousing and OLAP (DOLAP 2000)*, November 10, 2000, McLean, VA, USA, 33-40. ACM.

Pippal, S., Sharma, V., Mishra, S., & Kushwaha, D. S. (2011), "An Efficient Schema Shared Approach for Cloud based Multitenant Database with Authentication & Authorization Framework", *IEEE Computer Society*, p. 213-218.

Qian, L., Luo, Z., Du, Y. and Guo, L. (2009). "Cloud Computing: An Overview". Jaatun, M. G., Zhao, G. and Rong, C. (Eds.). LNCS 5931, p. 626–631, Berlin: Springer-Verlag.

Ramgovind, S., Eloff, M. M. and Smith, E. (2010). "The Management of Security in Cloud Computing". *IEEE*, p. 1-7.

Ranganathan, V. (2010). "Privacy Issues with Cloud Applications". *iS Channel*, Vol. 5 (1): p. 16-20. London School of Economics and Political Science.

Rosenthal, A. and Sciore, E. (2000). "View Security as the Basis for Data Warehouse Security", In: *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)*, M. Jeusfeld, H. Shu, M. Staudt, G. Vossen (eds.), Stockholm, Sweden, June 5-6, 1-8.

Ross, M. (2005). "The Matrix Revisited". *Intelligent Enterprise*, Vol. 8 (12): 42-44. Pro-Quest Computing.

Reese G 2009, Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, CA: O'Reilly Media, Inc.

Ruan, K., Carthy, J., Kechadi, T. and Crosbie, M. (2011). "Cloud Forensics: An Overview". Centre for Cybercrime Investigation, University College Dublin, 1-16.

Ruiter, J. and Warnier, M. (2011). “Privacy Regulations for Cloud Computing: Compliance and Implementation in Theory and Practice”. p. 355-389. Book Chapter: *Computers, Privacy and Data Protection: An Element of Choice*. Gutwirth, S., Poullet, Y., De Hert, P., Leenes, R. (Eds.). 1st edition. Springer.

Sabahi, F. (2011). "Virtualization-Level Security in Cloud Computing". *IEEE Computer Society*, p. 250-255. IEEE.

Sahlin, J., Thomas, S., and Mazzuchi, T. “Optimizing QoS in Distributed Systems/Cloud Computing Architecture”, International Journal Of Computer applications(0975-8887), PP 14-20, Volume 42, March 2012.

Schiefer, J. List, B., and Bruckner, R. M. (2002). "A holistic approach for managing requirements of data warehouse systems", IBM Watson Research Center in collaboration with Vienna University of Technology, In: *report of proceedings of eighth Americas conference on Information systems*: 77-87.

Sharma, R. and Sood, M. (2011). “Cloud SaaS and Model Driven Architecture”. RG Education Society, New Delhi, India, p. 16-23. IETE.

Shen, Z. and Tong, Q. (2010). "The Security of Cloud Computing System enabled by Trusted Computing Technology". *IEEE*, p. 11-15.

Sotomayor, B., Montero, R.S., Llorente, I. M. and Foster, I. (2009). "Virtual Infrastructure Management in Private and Hybrid Clouds". *IEEE Internet Computing, p. 14-22. IEEE.*

Stipic, A. and Bronzin, T. (2012). "How Cloud Computing Is (not) Changing the Way We Do BI", In MIPRO 2012, May 21-25,2012, Opatija, Croatia, 1574-1582, IEEE.

Stobla, N., Banek, M., and Tjoa, A. M. (2005). "The Security Issue of Federated Data Warehouses in the Area of Evidence-Based Medicine", Vienna University of Technology and University of Zagreb, 1-11.

Stoneburner, G., Gouguen, A. and Feringa, A. (2002). "Risk Management Guide for Information Technology Systems". Special Publication 800-30, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, p. 3-55.

Takabi, H., Joshi, J. B. D. and Ahn, G. (2010). "Security and Privacy Challenges in Cloud Computing Environments". *The IEEE Computer and Reliability Societies*, p. 24-31, IEEE.

Vrdoljak, B., Banek, M., and Rizzi, S. (2003). "Designing Web Warehouses from XML Schemas". *LNCS 2737*, Kambayashi, Y., Mohania, M. and Wob, W. (Eds). Springer-Verlag Berlin Heidelberg.

Xu, J., Zhang, D., Liu, L., & Li, X. (2012), "Dynamic Authentication for Cross-Realm SOA-Based Business Processes", *IEEE Transactions on services computing*, Vol. 5 (1), p. 20-32.

Wang, J., Wan, J., Liu, Z., and Wang, P. (2010). "Data Mining of Mass Storage based on Cloud Computing". *IEEE Computer Society*: 426-431. IEEE.

Wang, L., Jajodia, S., and Wijesekera, D. (2004). "Securing OLAP Data Cubes Against Privacy Breaches", In: *Proceedings of the 2004 IEEE Symposium on Security and Privacy (S&P'04)*, 1-15, IEEE Computer Society.

Wang, Q., Wang, C., Ren, K., Lou, W. and Li, J. (2011). "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing". *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22 (5): p. 847-859. IEEE.

Wohlin, C. . Runeson, P, M. Höst, M. C. Ohlsson, B. Regnell and A. Wesslén (2012). Experimentation in software engineering. Berlin: Springer.

Wan, Z. (2011). "A Network Virtualization Approach in Many-core Processor Based Cloud Computing Environment". *IEEE Computer Society*, p. 304-307. IEEE.

Wen, F. and Xiang, L. (2011). "The Study on Data Security in Cloud Computing based on Virtualization". *IEEE Computer Society*, p. 257-261. IEEE.

Wu, B. and Qin, L. (2011). "Design and Implementation of business-driven BI platform based on cloud computing", *IEEE Computer Society*: 118-122. IEEE.

Younge, A. J., Laszewski, G. V., Wang, L., Lopez-Alarcon, S., Carithers, W. (2010), "Efficient Resource Management for Cloud Computing Environments". *IEEE Computer Society*, p. 5. IEEE.

Zhang, D., & Xu, J. (2004, May). Multi-party authentication for web services: Protocols, implementation and evaluation. In *Object-Oriented Real-Time Distributed Computing, 2004. Proceedings. Seventh IEEE International Symposium on* (pp. 227-234). IEEE.

Appendix A: Experimental configuration in Eclipse

1. Configuration

- Sac

Sac consists of "**sac.jar"** file and "**certs**" folder.

There are **cloud folder**s in "certs". i.e**, "cloudA", "cloudB"** and so forth.

The cloud folder includes its **root key file**(i.e, "**cloudA.key**") and **uer folder**s(i.e, "**user1**").

The user folder has a **user key file.** (i.e, "**user1.key**")

The size of every key file is all **1024 Bytes.**

Example:

[SAC]

|_____ sac.jar

|____ [cloudA]

| |_____[user1]

| ||____user1.key

| |_____[user2]

| ||____user2.key

| |_____cloudA.key

|

|____ [cloudB]

… …

- Cloud

Cloud consists of "**cloud.jar**" file and **"cert"** folder.

The folder **"cert"** has one **user key file** - "**key.dat**" and one **ID file** - "**ID.dat**".

The **size** of key file **"key.dat"** is 2048 Bytes, where the first 1024 Bytes block is cloud root key and the next 1024 Bytes block is sub-domain key.

The **ID.dat** has two lines in it, where the first line is **cloud ID** (i.e, **cloudA**) and the next is **sub-domain ID**(that is, the user' s ID. i.e, **user1**). The cloud ID matches with the name of cloud folder, and the other with the name of user foloder.

The first block matches with the cloud root key in **SAC**, and the second with user key.

Example:

[user1]

|_____ cloud.jar

|____ [cert]

|____ID.dat

|____key.dat

- Service

Services consist of only one file **"service.jar"**

Well, so far we can understand the folder trees of each part.

2. Running and Testing protocols

See the folder "bin" and run programs in it.

- Running

- Firstly, run the sac.jar **once**, and turn on the SAC service state.
- Next, run the service **several times** (let us assume each instance is a different serv ice.) and give SAC IP addresses, service names and service ports. Then click "**Co**

nnect". For each run, give service names and ports **differently**.

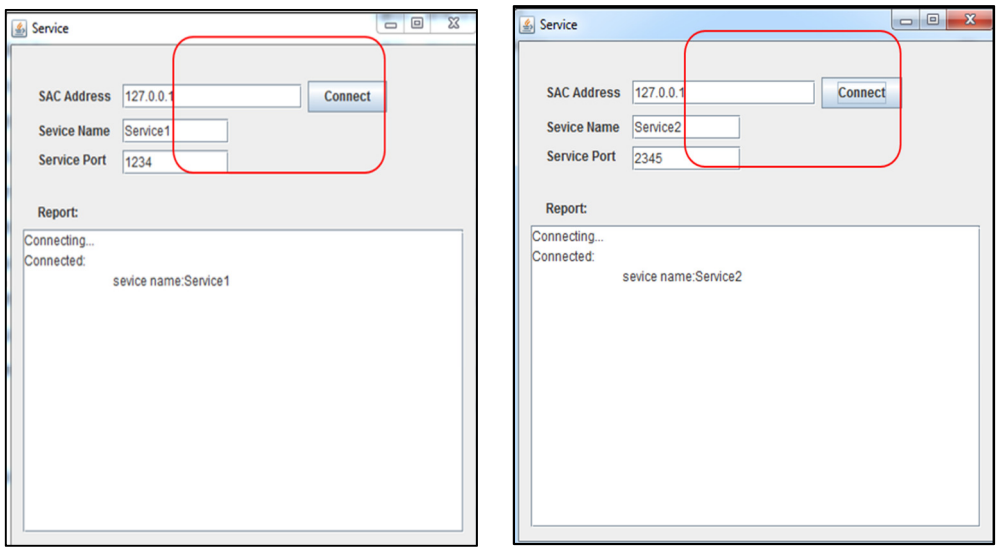

Figure 66: Cloud users connecting to multiple services

- Then, look at **SAC,** you can see that services are registered to SAC by clicking All "services" tab.

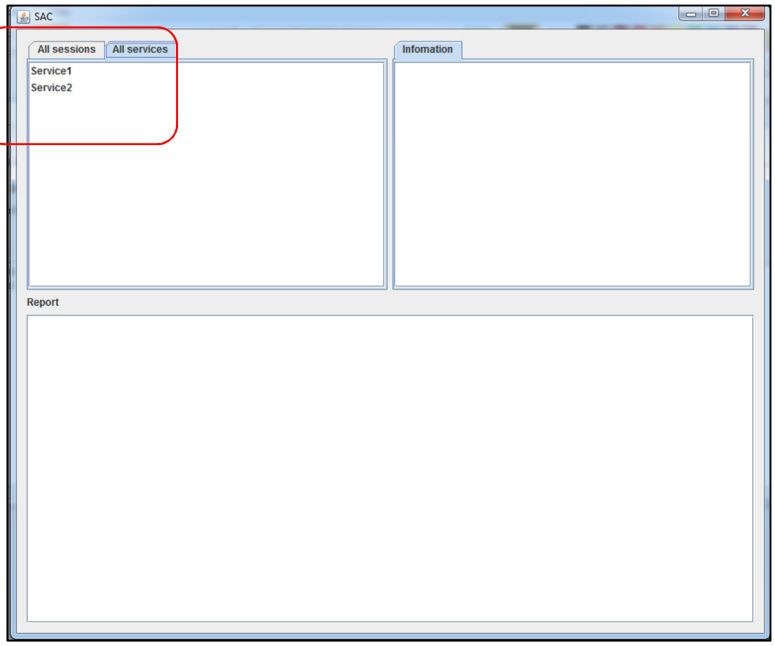

Figure 67: SAC services state

- Next, run cloud. Give SAC IP address and click "**Connect**". Then the services are listed

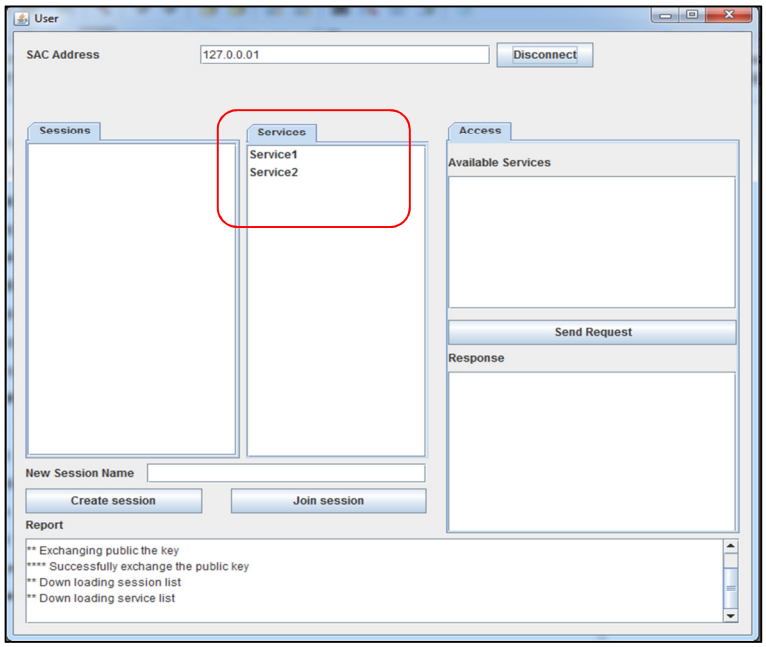

Figure 68: Accessing a SAC services after exchanging public the key

- ***Create session***

To create session, select services from the service list you want and give session name and then click "**Create session**". If the keys are valid, a new session is created and the available services are listed.

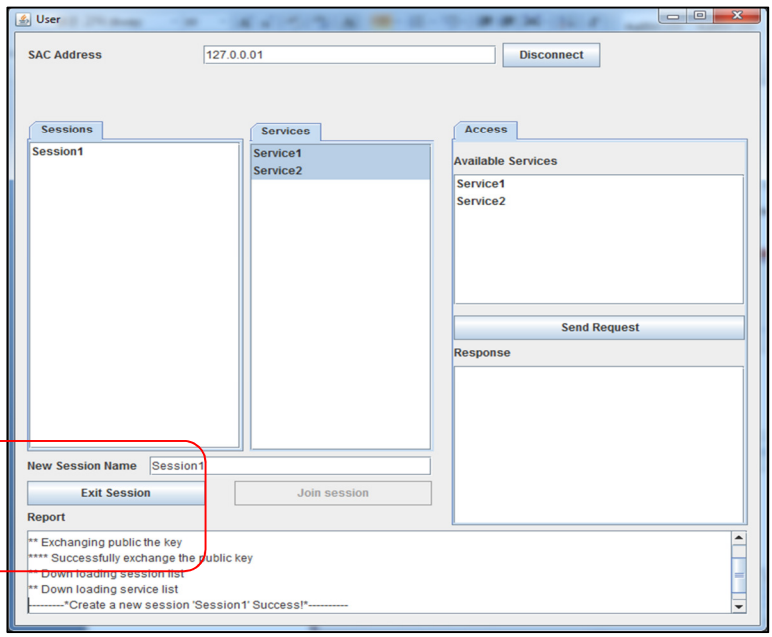

Figure 69: Creating a new session

If want to send request to a service, select a service from this list and click **"Send Request".** Then the response of the service will be displayed.

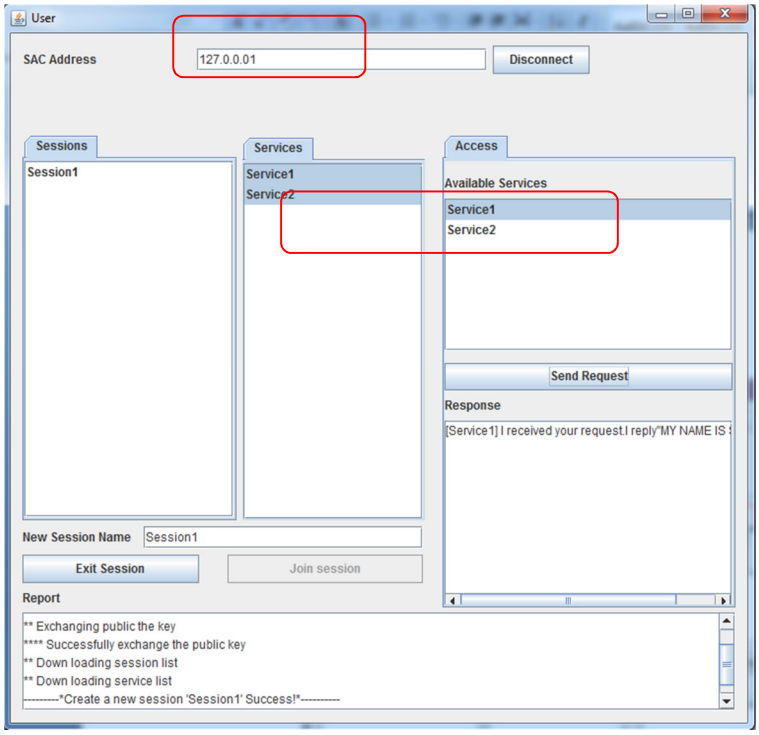

Figure 70: Cloud services request

See the SAC. You can see the information about sessions and services.

If want to exit session, click "**Exit session**". Then the session created will be disappeared in the session list.

- Join to session

  First run another "**cloud**". And connect to SAC.

  To join to a session, select a session from the list and click "**Join Session**".

  Then the cloud that created the session(**session creator**) will show a CONFIRM dialog. If the CREATOR wants to accept, click "**YES**". If not, "**NO**".

  According to it, the new user(cloud) may be allowed or not to join.

  If allowed, the list of services in the session will be displayed.

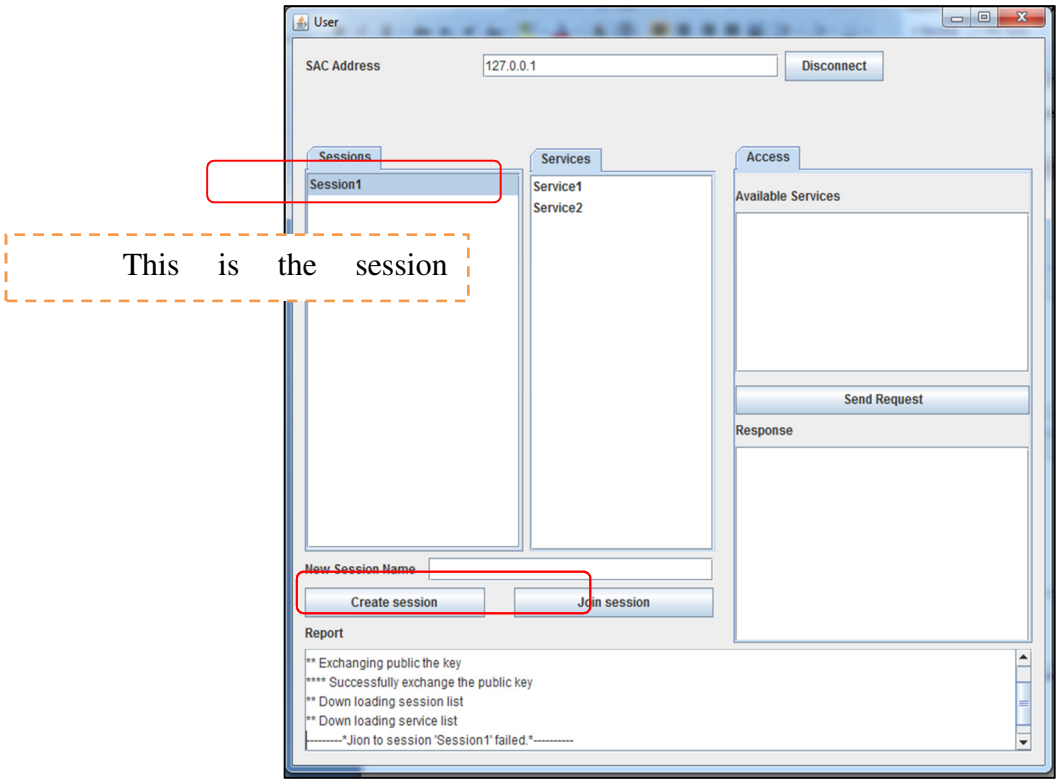

Figure 71: Joining a session

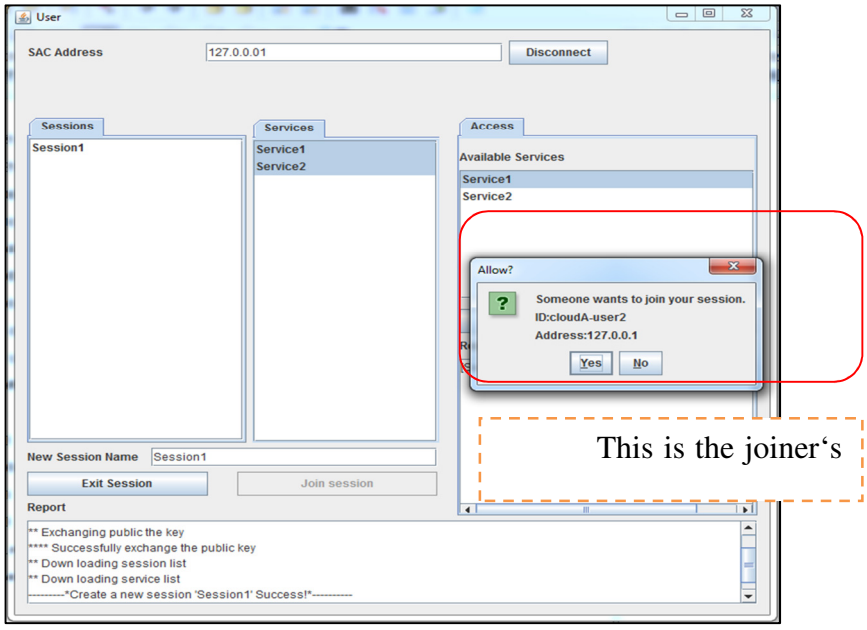

Figure 72: Accept or deny a joining session by Cloud user

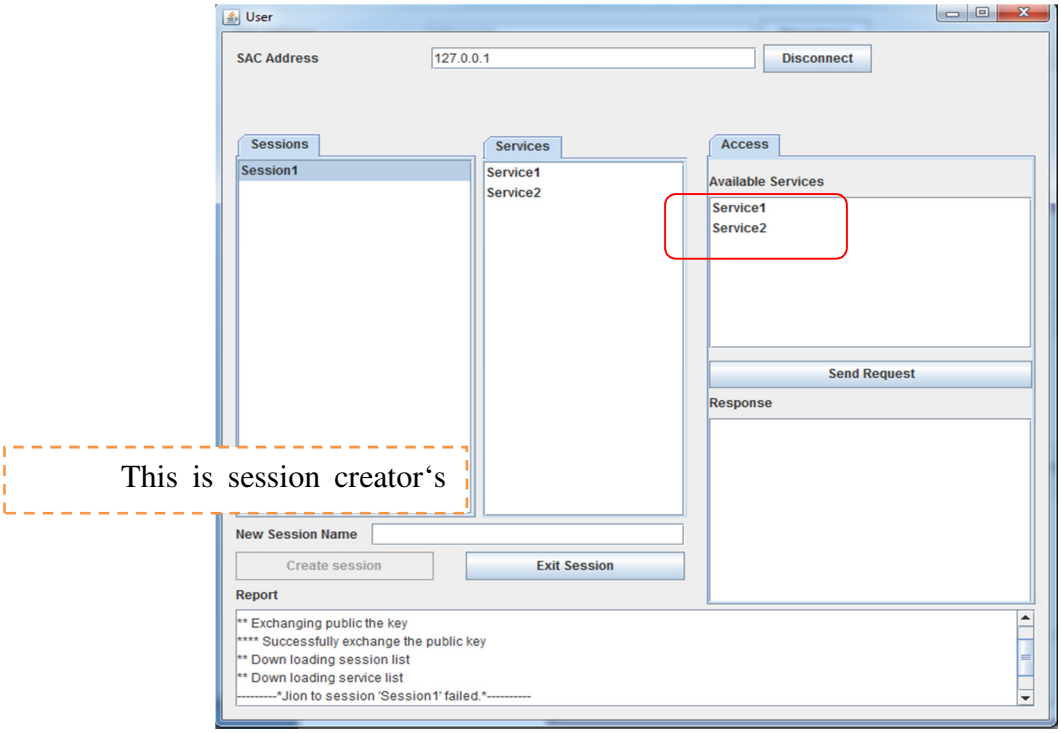

Figure 73: Creating a session

- In SAC you can see the info of sessions and services.

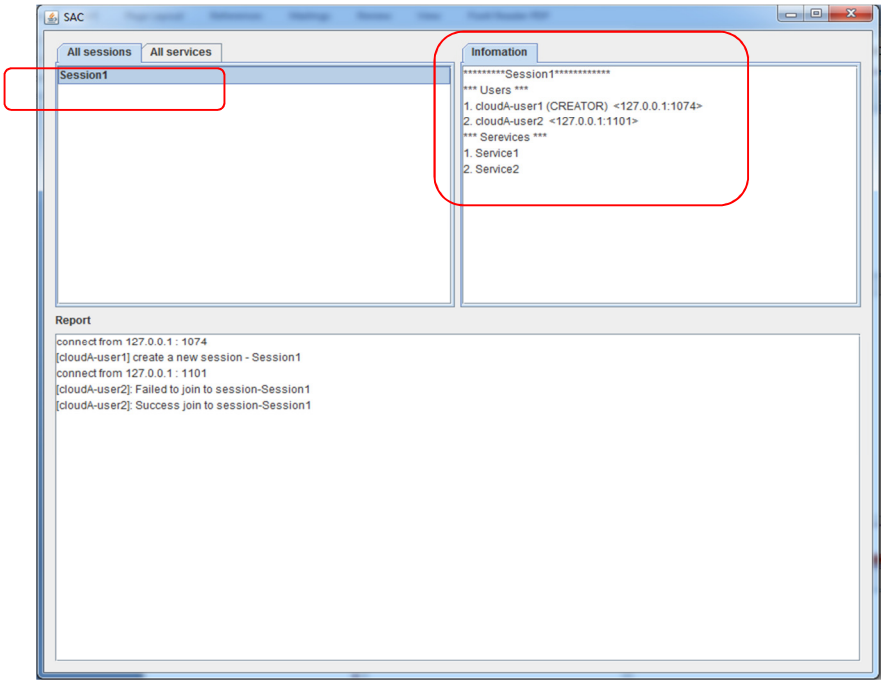

Figure 74: Session created by Cloud user (Creator)

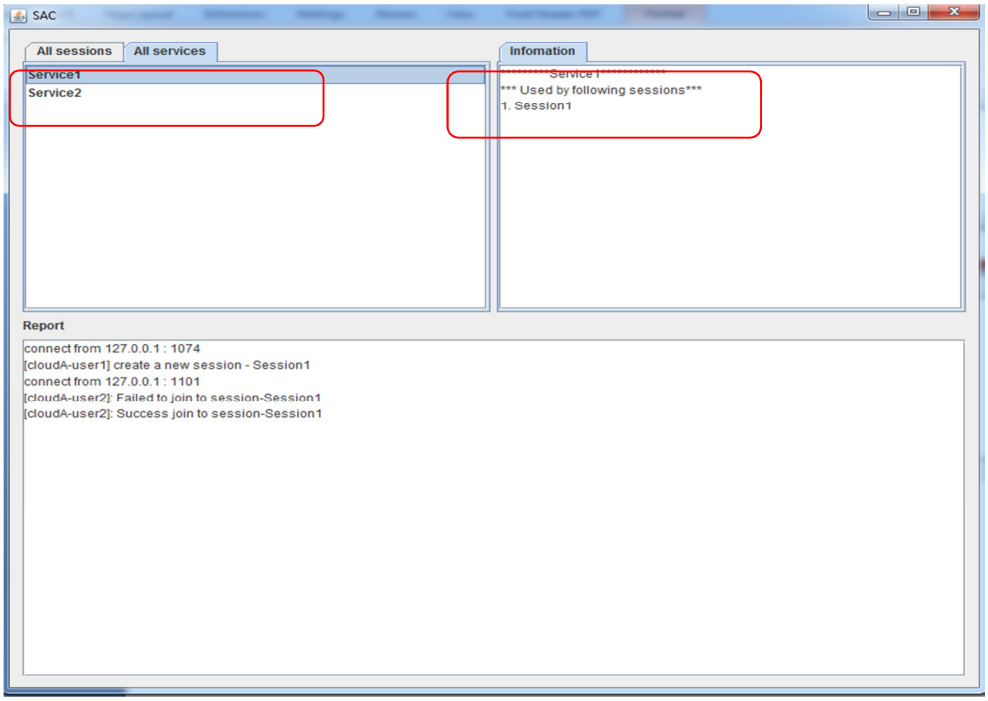

Figure 75: SAC services and sessions state

Appendix B: Experimental configuration in OPNET

Table 8: Application, database and security services modelled in OPNET for applying to server objects in the cloud hosted BI group and the UTM

Attribute	Value
Name	Profile
Model	Profile Configuration
ACE Tier Information	
Profile Definitions	
- Number of Rows	9
- Row 0	OLAP_Dashboards
- Row 1	DW_DW
- Row 2	UTM_DB-ACT
- Row 3	OLAP_Views
- Row 4	Antimalware
- Row 5	WEB_Security
- Row 6	Antispam
- Row 7	IDPS
- Row 8	LDAP

Table 9: Profiles created in the OPNET model for BI security on the cloud

Attribute	Value
Name	Profile
Model	Profile Configuration
Profile Configuration	
- Number of Rows	3
- Row 0	
- Profile Name	BI_Security_UTM
Applications	
- Operation Mode	Simultaneous
- Start Time (seconds)	Uniform (65,70)
- Duration (seconds)	End of Simulation
Repeatability	
- Row 1	
Applications	
- Profile_Name	BI_Application
- Operation Mode	Simultaneous
- Start Time (seconds)	Uniform (50,55)
- Duration (seconds)	End of Simulation
Repeatability	
- Row 2	
- Profile_Name	BI_DW_DW
Applications	
- Profile_Name	BI_Application
- Operation Mode	Simultaneous
- Start Time (seconds)	Uniform (50,60)
- Duration (seconds)	End of Simulation

Table 10: Creating the two applications–Protocol_Tasks (for all nodes), and database (for SAC-DB)

Attribute	Value
Name	Profile
Profile Definitions	
- Number of Rows	2
Protocol_Tasks	
- Profile Name	Protocol_Tasks
Description	
- Custom	Off
- Database	Off
- Email	Off
- Ftp	Off
- Http	Off
- Print	Off
- Remote Login	Off
- Video Conferencing	Off
- Voice	Off
SAC-DB	
- Name	SAC-DB
- Description	
- Custom	Off
- Database	High Load
- Email	Off
- Ftp	Off
- Http	Off
- Print	Off
- Remote Login	Off
- Video Conferencing	Off
- Voice	Off

Table 11: Destination preferences of A and F

Type	LAN
Attribute	Value
Name	A
Applications	
Applications ACE Tier Configuration	Unspecified
Application Destination Preferences	
- Number of Rows	2
Protocol_Tasks	
- Application	Protocol_Tasks
- Symbolic Name	F
- Actual Name	
- Number of Rows	1
- Enterprise Network F	
Protocol_Tasks	
Application Source Preferences	
Application Supported Profiles	
- Number of Rows	
Protocol_Tasks	
- Profile Name	Protocol_Tasks
- Number of Clients	Entire LAN
- Traffic Type	All Discrete
- Application Delay Tracking	Disabled
- Application Supported Services	None
LAN	
- LAN Background Utilisation	None
- LAN Server Name	Auto Assigned
- Number of Workstations	1000
- Switching Speed	500,000,000,000

Type	Server
Attribute	Value
Name	F
Applications	
Applications ACE Tier Configuration	Unspecified
Application Destination Preferences	
- Number of Rows	3
Protocol_Tasks	
- Application	Protocol_Tasks
- Symbolic Name	A
- Actual Name	
- Number of Rows	1
- Enterprise Network A	
- Name	Enterprise Network A
- Selection Weight	10
Protocol_Tasks	
- Profile Name	Protocol_Tasks
- Application	SAC
- Symbolic Name	F
- Actual Name	
Application Multicasting Specification	None
Application RSVP Parameters	None
- Application Segment Size	64,000
Application Source Preferences	None
Application Supported Profiles	
- Number of Rows	1
Protocol_Tasks	Protocol_Tasks
- Profile Name	All Discrete
- Traffic Type	Disabled
- Application Supported Services	None

Table 12: Destination preferences of SAC-SH

Type	Server
Attribute	Value
Application Destination Preferences	
- Number of Rows	3
Protocol_Tasks	
- Application	Protocol_Tasks
- Symbolic Name	CloudA
- Actual Name	
- Number of Rows	4
- Enterprise Network CloudA	...
- Enterprise Network CloudA_1	...
- Enterprise Network CloudA_2	...
- Enterprise Network CloudA_3	...
Protocol_Tasks	
- Application	Protocol_Tasks
- Symbolic Name	CloudB
- Actual Name	
- Number of Rows	4
- Enterprise Network CloudB	...
- Enterprise Network CloudB_1	...
- Enterprise Network CloudB_2	...
- Enterprise Network CloudB_3	...